

令和元年度 修士論文

時空間の多様化・集中化に基づく
適応型 Cuckoo Search

首都大学東京大学院 システムデザイン研究科
電子情報システム工学域

曹 洋

論文要旨

最適化は工学，経済，金融などの様々な分野で応用されている。そして，社会の発展に対して重要な位置となっている。近年のシステムと工業製品は段々大規模化と複雑化になっていて，システムの設計・運用・制御に対して一層の高効率化・高信頼化・高機能化の実現が重要な課題となっている。また，シミュレーションとモデリング技術の発展のとともに，最適化を行う環境が急速に発展している。実用の最適化のニーズに対しては，新たな最適化の枠組みの構築や最適化手法の開発のニーズが必要となる。そして，このニーズに対して決定変数情報と評価値情報だけで探索が可能であり，高い実用性を有するメタヒューリスティクスは注目されている。メタヒューリスティクスは解情報と対象の評価値情報だけ利用して最適化を行う手法であり，時間内でユーザの要求に応じて近似解を求めるという特徴がある。一方，メタヒューリスティクスは調整可能な様々なパラメータがあり，問題に応じて適切なパラメータを調整できれば幅広い問題に効率的に解決できる。様々な問題を効率的に解くためには，様々な問題の構造によって適応的なパラメータ調整機能を有することが必要である。以上から，メタヒューリスティクスは様々な実システムに対して効率的に最適化を行えるために，適応的なパラメータの調整則を開発する課題がある。

先行研究はメタヒューリスティクスの以上の課題を解決するために，パラメータの多様性と自由度の拡張，また適切なパラメータの調整機能の付加の重要性を踏まえ，近接最適性原理（POP）と空間的な多様化・集中化の二つの戦略を用いて多点探索型メタヒューリスティクスの一つである Cuckoo Search（CS）の探索性能を向上した。先行研究では CS の探索構造（近傍生成と更新機構）に着目し，CS の特徴を活かすことで改良を実現している。具体的に先行研究では，CS の他の個体の情報だけ用いて近傍解を生成する特徴と，パラメータによる近傍生成の範囲を調整できる特徴を活かし，CS の各探索点のパラメータを

多様化するとともに、各探索点のランキングに基づく適応的なパラメータ調整機能を付加した適応型 CS を提案した。

先行研究においては CS の適応性が向上したが、時間軸から見ると多様化・集中化に基づく制御が一様、つまり毎回の探索の状態によりパラメータ調整則が一様になっているため、時間的な多様化・集中化は十分には実現できていない。言い換えれば、先行研究では多様化・集中化の実現が解くべき問題の種類や次元、初期値や反復回数に依存し、探索状態の制御が十分に行えないという課題が残されている。このことは、探索序盤には探索点は空間的には集中化するが時間軸的には多様化すべきであり、また、探索終盤には探索点は空間的には多様化するが、時間軸的には集中化させる戦略が必要であることを意味する。

本研究は先行研究における上述の課題を解決するために、CS の新たなパラメータ調整戦略として、CS の近傍生成に対して先行研究の探索戦略の上で時間的な多様化・集中化機能を追加している。本研究で提案する新たな CS は、時間的な多様化・集中化戦略として探索序盤では探索点はより広い範囲で探索することで多様化を実現して長期的な解の改善を実現する一方で、探索終盤では良い解付近を集中的に探索することで短期的な解の改善を実現している。本研究は CS の近傍生成に対しては空間的な多様化・集中化を実現するために、空間的（各探索点の評価値のランキングに応じた）な多様化・集中化の戦略と、時間的な多様化・集中化戦略を有機的に結合した新たな CS の近傍生成方法を提案した。本研究は探索の反復回数に応じて多様化・集中化が十分に実現できるように、調整すべきパラメータ β の上下限がスケジュールとして設定されている。本研究を通じて先行研究の CS の近傍生成の時間的な多様化・集中化が不十分であるという課題を解決した。本研究では、探索のイテレーション回数に応じた適応的な多様化・集中化を実現することができる。言い換えれば、本研究を通して探索のイテレーション回数を問わずに、探索初期の十分の多様化と探索終盤の十分の集中化を確実に保証できる。実験を通して、排斥操作がない場合でも、CS の近傍生成に時間的な多様化・集中化を追加した提案手法は、先行研究の CS より探索性能が優れていること、およびベンチマーク関数の種類、次元数、初期値などを問わずに、高いレベルの多様化・集中化を実現できることを検証した。

また、先行研究では CS の排斥率をゼロとしたが、本研究では排斥手法の改良も行った。本研究では排斥の空間的な多様化・集中化を実現するために、まず排斥する点の選点方式に対して改良した。従来の CS は排斥する時に評価値が一番悪い探索点しか排斥点として排斥していないが、適応性が低い。本研究ではルーレット選択方法を導入して、探索点の

適応度比率によって排斥点を選択して排斥する。そして、近傍生成の探索戦略との一貫性を踏まえ、選択した排斥点の評価値に応じて CS の排斥の操作には空間的な多様化・集中化と時間的な多様化・集中化の探索戦略を同時に活かし、新たな排斥の調整則も提案した。最後に、典型的なベンチマーク問題を用いた数値実験を通じて、提案する新たな適応型 CS の探索性能を詳細に検証・評価した。

目次

論文要旨	i
1 序論	1
1.1 論文の背景	1
1.2 本論文の構成	4
2 代表的なメタヒューリスティクスと探索戦略	5
2.1 代表的なメタヒューリスティクスの紹介	5
2.1.1 Particle Swarm Optimization	5
2.1.2 Differential Evolution	7
2.1.3 Cuckoo Search	8
2.2 探索戦略	12
2.2.1 近接最適性原理	12
2.2.2 多様化・集中化	13
3 Cuckoo Search における先行研究と課題	15
3.1 先行研究の位置付け	15
3.2 Cuckoo Search のパラメータ解析と調整	16
3.2.1 Cuckoo Search のパラメータ解析	16
3.2.2 Cuckoo Search のパラメータ調整	17
3.3 探索点のランキングに基づく先行研究の提案	18
3.4 先行研究の中に残った課題	20

4	時空間の多様化・集中化に基づく適応型	
	Cuckoo Search	24
4.1	本研究の位置付け	24
4.2	CS の近傍生成における時空間の多様化・集中化の実現	26
4.3	CS の排斥における時空間の多様化・集中化の実現	27
4.4	数値実験	30
4.4.1	数値実験条件	30
4.4.2	実験結果・考察	30
5	結論	43
5.1	まとめ	43
5.2	展望	45
	参考文献	46
A	ベンチマーク関数	48
	謝辞	52

1

序論

1.1 論文の背景

最適化は工学，経済，金融などの様々な分野で応用されている。そして，社会の発展に対して重要な役割を果たしている。工学に対して重要な位置を持っている最適化は，目的関数 $f(\mathbf{x}) \in \mathbb{R}$ の最小解・最大解 $\mathbf{x}^* \in \mathbb{R}^n$ を求める関数の最適化問題である。また，近年のシステムと工業製品は段々大規模化と複雑化になっていて，システムの設計・運用・制御における一層の高効率化・高信頼化・高機能化の実現が重要な課題となっている。また，シミュレーションとモデリング技術の発展とともに，最適化を行う環境が急速に発展している。実用の最適化の発展に対しては，新たな最適化の枠組みの構築や最適化手法の開発のニーズが必要となる。実用的な最適化のニーズに対しては，新たな最適化の枠組みの構築や最適化手法の開発が必要なる。

最適化分野の一つである手法の数値計画法は，1970年代の高速・大容量のコンピュータの計算パワーの増大とともに，大きな発展を取った。代表的な数値計画法は最急降下法，準ニュートン法などがある[1, 2]。しかし，当時の工学分野における最適化技術の発展がまだ遅いので，数値計画法はモデリングが容易な工学分野に限定されていた。複雑化・大規模になっているシステムの設計・解析・制御における最適化に対するニーズは高めているが，数値計画法を使う時に，微分可能性と連続性などの条件を満足するモデリングが必要となる。そのため，システムから得たモデルと実システムの差が大きくなり，求めた解の実用性や最適性が悪いという課題がある。シミュレータなどの関連技術が最適化

において十分には活用されていない。そして、数理計画法はシステムの複雑化・大規模化に従って必要な計算量はかなり増大し、実システムへの適用はかなり困難である[1]。以上から見ると、コンピュータ技術の急速な発展や最適化技術の進歩など、最適化を行う環境やニーズの変化に従い、新たな最適化の枠組みの構築や実用性が高い最適化アルゴリズムの開発が重要な課題となっている。

最適化の実用性を高めるために、新たな最適化アルゴリズムの開発が必要なるが、近年決定変数情報と評価値情報だけで探索が可能であり、高い実用性を有するメタヒューリスティクスは注目されている。代表的なメタヒューリスティクス手法は遺伝的なアルゴリズム (GA), Differential Evolution (DE), タブサーチ (TS), Particle Swarm Optimization (PSO) [1, 3] などがある。メタヒューリスティクスの特徴の一つとしては、物理現象、生物現象、あるいは生物の集団・社会現象などにアナロジーを持つものが多い。メタヒューリスティクスは従来の数理計画法に比べて実用性が高い。メタヒューリスティクスは微分可能性や連続性などの条件が必要なく、解情報と対象の評価値情報だけ利用して最適化を行うことが出来て、時間内でユーザの要求に応じて近似解を求めるという特徴がある。多くのメタヒューリスティクスは、複数の探索点を有する多点探索型であるが、多点探索型メタヒューリスティクスにおいては、探索点が全て一様なパラメータで探索を行うことが一般的である[1, 7, 10]。もし一様ではないパラメータで探索を行えば、メタヒューリスティクスの探索性能を向上することが期待できる。一方、もしアルゴリズムに適切なパラメータの調整機能が付加されなかったら、アルゴリズムの探索性能が悪化する可能性もある。

先行研究は上述の多点探索型メタヒューリスティクスにおけるパラメータ設定の自由度の拡張と、適切なパラメータの調整機能の付加の両立の重要性を踏まえ、近接最適性原理 (POP) と空間的な多様化・集中化の二つの戦略を用いて多点探索型メタヒューリスティクスである Cuckoo Search (CS) [8, 9] の探索性能を向上した。先行研究は CS のパラメータの自由度と適応性を向上して空間的な多様化・集中化を実現した。メタヒューリスティクスの一手法であり、鳥のカッコウの繁殖行動を模擬した Cuckoo Search (CS) は一般的なメタヒューリスティクス手法と違って他の探索点の情報だけで近傍解を生成している。そして、CS はレヴィ分布で近傍解を生成し、近傍生成範囲はパラメータで調整することができる。先行研究では CS の近傍生成と更新機構に着目し、CS の特徴を活かすことで改良を実現している。具体的に CS の他の個体の情報だけ用いて近傍解を生成する特徴と、パラメータにより近傍生成の範囲を調整できる特徴を活かし、CS の各探索点のパラメータを多

様化するとともに、各探索点のランキングに基づく適応的なパラメータ調整機能を付加した適応型 CS を提案した。

先行研究では CS の適応性が向上したが、時間軸から見ると多様化・集中化に基づく制御が同じ、つまり毎回の探索の状態によりパラメータ調整則が同じになっている。そのため、先行研究は反復回数の変化に伴い、探索状態の制御が十分ではないという課題が残している。具体的に、探索序盤には探索点は空間的には集中化するが時間軸的には多様化し、また、探索終盤には探索点は空間的には多様化するが、時間軸的には集中化させる戦略が必要である。先行研究において提案された空間的な多様化・集中化でも、ある程度の時間的な多様化・集中化を実現できるが、先行研究の多様化・集中化は、探索点の初期値、ベンチマーク関数の種類、探索点数などからの影響で十分な多様化・集中化を実現できない。先行研究の多様化・集中化のレベルまだ低い。先行研究では、探索終了する時に、集中化できないことがあるし、探索のイテレーション回数の半分ぐらいもう強く集中化して、多様化が足りないこともある。つまり、先行研究では、多様化・集中化の制御がまだ不十分だという課題がある。

本研究は先行研究のパラメータ調整則が一様になっている課題を解決するために、CS の新たなパラメータ調整戦略として、時間的な多様化・集中化機能を追加している。さらに、CS の排斥の操作には先行研究の探索戦略と本研究の探索戦略を同時に活かすことを可能とする新たな排斥の調整則を提案した。本研究で提案する新たな CS は、時間的な多様化・集中化戦略として探索序盤では探索点はより広い範囲で探索することで多様化を実現して長期的な解の改善を実現する一方で、探索終盤では良い解付近を集中的に探索することで短期的な解の改善を実現している。

具体的に本研究は CS の近傍生成に対しては空間的な多様化・集中化を実現するために、各探索点の評価値に応じてランキングに基づいてパラメータを調整する戦略と、時間的な多様化・集中化戦略を適切に結合した新たな CS の近傍生成方法を提案し、先行研究の CS の近傍生成の時間的な多様化・集中化が不十分という課題を解決した。本研究では、探索のイテレーション回数に応じた適応的な多様化・集中化を実現することができる。言い換えれば、本研究を通して探索のイテレーション回数を問わずに、探索初期の十分な多様化と探索終盤の十分な集中化を確実に保証できることを意味する。実験を通して、排斥がない場合でも、CS の近傍生成に時間的な多様化・集中化を追加することで先行研究の CS の性能を上回ることを検証した。

さらに、本研究では排斥の空間的な多様化・集中化を実現するために、まず排斥する点の選点方式に対して改良した。従来のCSは排斥する時に評価値が一番悪い点は排斥点として排斥している。適応性が低い。本研究ではルーレット選択方法を導入して、探索点の適応比率によって排斥点を選択して排斥する。そして、選択した排斥点の評価値に応じてランキングに基づいてパラメータを調整する戦略と、時間的な多様化・集中化戦略を有機的に結合して、新たなCSの排斥手法も提案した。排斥手法においても時空間の最適化を実現した。この新たな排斥に関する改良を通して、さらにCSの高次元の探索性能の向上を実現した。

最後、本研究は典型的なベンチマーク問題を用いた数値実験により、オリジナルCSおよび先行研究と比較しつつ、提案手法の探索性能を詳細に検証・評価した。

1.2 本論文の構成

本論文は全5章から構成されている。各章の概要を以下に示す。

第1章の「序論」では、本論文の背景と目的を述べる。

第2章の「メタヒューリスティクスと探索戦略」では、代表的なメタヒューリスティクス手法と本研究で採用したCuckoo Searchについて紹介する。本研究のメタヒューリスティクスの手法の特徴を挙げる。さらに、選択した手法の特徴により使った探索戦略を紹介する。

第3章の「先行研究」では、Cuckoo Searchの適応性とパラメータの自由度と多様性を向上し、CSの空間的な多様化・集中化を実現している先行研究について述べる。そして、先行研究の中にまだ存在している課題について分析する。

第4章の「時空間の多様化・集中化を基づく適応型Cuckoo Search」では、先行研究の中に残った課題を解決するために、行った本研究の位置付けと具体的な改良方法について紹介する。さらに、数値実験により本研究の有用性を説明する。

第5章の「結論」では、本論文のまとめと今後の課題について述べる。

2

代表的なメタヒューリスティクスと探索戦略

本章ではまず代表的なメタヒューリスティクスについて紹介する。そして、本研究の手法 Cuckoo Search を選択した理由について説明する。また、Cuckoo Search の特徴に合わせて使った探索戦略について紹介する。

2.1 代表的なメタヒューリスティクスの紹介

2.1.1 Particle Swarm Optimization

Particle Swarm Optimization(PSO) アルゴリズムは、社会モデルから発想し、1995 年において開発されたメタヒューリスティクスの一つである[10]。Particle Swarm Optimization の中の Particle（探索粒子）は位置と速度を持っている微小粒子を真似している探索点である。Swarm（群れ）は人工生命の研究における群知能の中の群れである。

PSO アルゴリズムはメタヒューリスティクスの一つの手法である。PSO は生物の群れが餌を探す行動から発想し、探索情報をすべて探索個体に共有している。群の中の個体がそれぞれ独立で行動することではなく、群の中の個体の探索情報と探索群の全体の共通情報を交換して、一定のルールで探索を行う。

メタヒューリスティクスの一つである PSO アルゴリズムの特徴は、1.探索点は複数であり、多点探索のメタヒューリスティクスである。2.探索個体はお互いに最良解に関する情報を共有して、探索を行う[10]。また、PSO はアルゴリズムとして構造が簡単であり、基本的

な算術演算だけで探索を行える。PSO の具体的なアルゴリズムを以下に示す。

【Particle Swarm Optimization】

Step 0:[準備]

探索点数 $2 \leq m \in \mathbb{N}^1$, 探索点のパラメータ $0 < w \in \mathbb{R}^1$, $0 < c_1 \in \mathbb{R}^1$, $0 < c_2 \in \mathbb{R}^1$, 最大反復回数 T_{\max} を設定, $k = 0$ とする。

Step 1:[初期化]

各探索点の初期位置 $\mathbf{x}_i^0 \in \mathbb{R}^n$ ($i = 1, 2, \dots, m$), 初期速度 $\mathbf{v}_i^0 \in \mathbb{R}^n$ ($i = 1, 2, \dots, m$) を与える。初期位置 \mathbf{x}_i^0 はランダムで生成し, 初期速度 \mathbf{v}_i^0 もランダムで生成する。

$$pbest_i^0 = \mathbf{x}_i^0 (i = 1, 2, \dots, m)$$

$$gbest^0 = \arg \min_{pbest_i^0} f(pbest_i^0)$$

とする。

Step 2:[$\mathbf{v}_i, \mathbf{x}_i$ の更新]

$$v_{ij}^{k+1} = w \cdot v_{ij}^k + c_1 \cdot rand_1()_{ij} \cdot (pbest_{ij}^k - x_{ij}^k) + c_2 \cdot rand_2()_{ij} \cdot (gbest_j^k - x_{ij}^k)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}$$

$$i = 1, 2, \dots, m; j = 1, 2, \dots, n$$

とおく。

Step 3:[$pbest_i, gbest$ の更新]

$$I = \{i | f(\mathbf{x}_i^{k+1}) < f(pbest_i^k), i = 1, 2, \dots, m\}$$

とおく。ここで, $i \in I$ ならば,

$$pbest_i^{k+1} = \mathbf{x}_i^{k+1}$$

もしなければ,

$$pbest_i^{k+1} = pbest_i^k$$

さらに,

$$gbest^{k+1} = \arg \min_{pbest_i^{k+1}} f(pbest_i^{k+1})$$

Step 4:[終了判定]

$k = T_{max}$ の時に探索終了する。もしなければ, $k := k + 1$ として Step 2 へ戻る。

上の式の中に記号の定義としては, $n \in \mathbf{N}^1$: 問題の次元, $m \in \mathbf{N}^1$: 探索点数, $\mathbf{x}_i \in \mathbf{R}^n$: i 番目の探索点の位置ベクトル, $\mathbf{v}_i \in \mathbf{R}^n$: i 番目の探索点の移動ベクトル, $f(\mathbf{x}_i) \in \mathbf{R}^1$: i 番目の探索点の評価値, $pbest_i \in \mathbf{R}^n$: i 番目の探索点がかこれまでの探索で発見した最良解, $gbest \in \mathbf{R}^n$: これまでの探索で発見した群全体の最良解, $w, c_1, c_2 \in \mathbf{R}^1$: PSO のパラメータ, $rand() \in \mathbf{R}^1$: 0 から 1 までの間に分布する一様乱数, $T_{max} \in \mathbf{N}^1$: イテレーションの上限回数。

2.1.2 Differential Evolution

Differential Evolution (DE) アルゴリズムは 1995 年において開発し, メタヒューリスティクスの一つである。遺伝的なアルゴリズム GA のように, 突然変異, 交叉, 選択の進化操作で探索を行うメタヒューリスティクス手法である[6]。一般的なメタヒューリスティクスと同じに様々な調整できるパラメータを持っている特徴がある。そして, パラメータの適切な調整でパラメータの自由度を向上出来て, アルゴリズムの探索の効率を向上することができる。一方で, ユーザはアルゴリズムを使用する時に, 様々なパラメータに対しての適切な調整は課題になっている。DE アルゴリズムの特徴として, 調整すべきパラメータが少なく, ユーザへのパラメータの設定する負担が少ない。Differential Evolution の具体的なアルゴリズムを以下に示す。

【Differential Evolution】

Step 0:[準備]

探索ベクトルの次元数 $n \in \mathbb{N}^1$, 探索ベクトルの個体数 $m \in \mathbb{N}^1$, 膨張率 $F \in \mathbb{R}^1$, 交叉率 $CR \in \mathbb{R}^1$, 最大世代数 $G_{\max} \in \mathbb{N}^1$ とする。

Step 1:[初期化]

初期解集団 $\mathbf{x}_i^0 \in \mathbb{R}^n$ ($i = 1, 2, \dots, m$) を与え, 世代数 $G = 0$ とする。

Step 2:[突然変異]

全操作ベクトル \mathbf{x}_i^G に対して, 互いに異なるようにランダムに選出された $\mathbf{x}_{r1}^G, \mathbf{x}_{r2}^G, \mathbf{x}_{r3}^G$ と膨張率 F を用いて, 下記式の変異ベクトル $\mathbf{v}_i^G \in \mathbb{R}^n$ を生成する。

$$\mathbf{v}_i^G = \mathbf{x}_{r1}^G + F \cdot (\mathbf{x}_{r2}^G - \mathbf{x}_{r3}^G)$$

Step 3:[交叉 (一様交叉)]

全操作ベクトル \mathbf{x}_i^G に対して, 交叉開始点 a を決めし, 操作ベクトルの要素 $x_{i,j}$ 毎に交叉率 CR を用いて交叉判定を行い, $x_{i,j}^G$ と $v_{i,j}^G$ の要素を次式のように置換し, 試験ベクトル $\mathbf{u}_i^G \in \mathbb{R}^n$ を生成する。

$$u_{i,j}^G = \begin{cases} v_{i,j}^G & r_j \leq CR \text{ or } j = a \\ x_{i,j}^G & \text{otherwise} \end{cases}$$

Step 4:[選択]

全操作ベクトル \mathbf{x}_i^G に対して, 次式のように \mathbf{u}_i^G との比較を行い, 優れる評価値を持つ解を \mathbf{x}_i^{G+1} とする。

$$\mathbf{x}_i^{G+1} = \begin{cases} \mathbf{u}_i^G & f(\mathbf{u}_i^G) \leq f(\mathbf{x}_i^G) \\ \mathbf{x}_i^G & \text{otherwise} \end{cases}$$

Step 5:[終了判定]

$G := G + 1$ とし, $G = G_{\max}$ であれば, 探索が終了する。もしなければ, Step 2 へ戻る。

2.1.3 Cuckoo Search

Cuckoo Search (CS) アルゴリズムは Xin-She Yang により 2009 年において開発された手法であり, 連続最適化問題を解決であり, 多点探索のメタヒューリスティクス手法の一

つである[8, 9]。Cuckoo Search は鳥のカッコウの托卵行動を真似している。鳥のカッコウは他の鳥の巣に卵を産み、卵と雛の世話は他の鳥で実施して托卵という習性がある。そして、卵を産み付けられる巣の親鳥を仮親という。カッコウは自身の卵が仮親自身の卵のように見せかける。また、仮親は自身以外の卵に気づいたら、その卵を自分の巣から捨てる。あるいはこの巣を捨てて新たな巣を作る。CS はカッコウの上述の行動によって開発した手法であり、主に以下の3つの操作：「レヴィフライト」、「更新」、および「排斥」によって探索を行う。

- (1) 各カッコウはランダムに巣を選び、そして、選んだ巣に一つの卵を産み付ける。
- (2) 良い卵がある巣は次の世代へ入る。
- (3) 仮親はある確率でカッコウは産み付けた卵を発見し、発見した卵を捨てて、新たな巣を作る。

レヴィフライトは、レヴィ分布に従い、ランダム生成の一種である。レヴィフライトのランダム生成の特徴により、広い範囲で効率的に探索を行える。また、いくつかのレヴィ分布に従う乱数の発生方法があるが、こちらは Mantegna's Algorithm (MA) の乱数発生方法を使う[11]。

分布調整変数 β によって、近似レヴィ分布 Le の形状は変わることができる。分布調整変数 β は $0.3 \leq \beta \leq 1.99$ のときにレヴィフライトがレヴィ分布を近似する。MA により $L(\beta)$ は生成される近似レヴィ分布に従う乱数を生成し、以下の式を用いる。

$$L(\beta) = \frac{u}{|v|^{1/\beta}} \quad (2.1)$$

v は標準正規分布に従う乱数である。

平均値 $u=0$ 、分散 σ_u の正規分布に従う乱数として式 (2.2) で与えられる。分散 σ_u は式 (2.3) で表される。

$$u \sim N(0, \sigma_u^2) \quad (2.2)$$

$$\sigma_u = \left[\frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2] \beta 2^{(\beta-1)/2}} \right]^{1/\beta} \quad (2.3)$$

$\Gamma(x)$ は以下のガンマ関数である。

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \quad (x > 0) \quad (2.4)$$

CSの探索はまずランダムに更新点と参照点 \mathbf{x}^p を選び、選んだ参照点によりレヴィフライイトで近傍解 $\hat{\mathbf{x}}$ を生成する。そして、更新点と近傍解を比較して、探索を行う。レヴィフライイトは、選択した参照点 \mathbf{x}^p に、ステップサイズ α とレヴィ分布に従う乱数 $L(\beta)$ を掛けた値を加えることで、近傍解を生成する操作です。式 (2.5) で示す。

$$\begin{aligned}\hat{\mathbf{x}} &:= \mathbf{x}^p + \alpha L(\beta), \\ L(\beta) &= [L_1(\beta), L_2(\beta), \dots, L_n(\beta)]^T\end{aligned}\tag{2.5}$$

ここで、 n は対象問題の次元数である。

更新では、参照点を抜いた探索点群からランダムに更新点 \mathbf{x}^q (更新点 $\mathbf{x}^q (\neq \mathbf{x}^p)$) を選ぶ。そして、更新点 \mathbf{x}^q の評価値と近傍解 $\hat{\mathbf{x}}$ の評価値と比較し、更新点の評価値が優れている場合に更新する。排斥では、排斥率 p_a に従って評価値が一番悪い個体 \mathbf{x}^w を排斥させる。式 (2.8) で示す。

$$\begin{aligned}\mathbf{x}^w &:= \mathbf{x}^w + \alpha L(\beta) \\ L(\beta) &= [L_1(\beta), L_2(\beta), \dots, L_n(\beta)]^T\end{aligned}\tag{2.6}$$

以下にオリジナルの Cuckoo Search のアルゴリズムを示す[8, 9]。

【Cuckoo Search】

Step 0:[準備]

探索点数 m , ステップサイズ $\alpha > 0$, 分布調整パラメータの設定範囲 $\beta \in [0.3, 1.99]$, 排斥率 $p_a \in [0, 1]$, 最大評価回数 k_{\max} を定め、評価回数を $k = 0$ とする。

Step 1:[初期化]

初期の探索点 \mathbf{x}^i ($i = 1, 2, \dots, m$) は初期配置領域 S 内からランダムに生成し、初期探索点群を $\mathbf{X} = \{\mathbf{x}^i \mid i = 1, 2, \dots, m\}$ とする。初期探索点の評価値 $f(\mathbf{x}^i)$ を求め、 $k := m$ とする。

Step 2:[レヴィフライイト]

ランダムに探索点群から参照点 $\mathbf{x}^p \in \mathbf{X}$ を選び、近傍解 $\hat{\mathbf{x}}$ を生成する。

$$\begin{aligned}\hat{\mathbf{x}} &:= \mathbf{x}^p + \alpha L(\beta) \\ L(\beta) &= [L_1(\beta), L_2(\beta), \dots, L_n(\beta)]^T\end{aligned}$$

近傍解の評価値 $f(\hat{x})$ を求め、 $k := k + 1$ とする。

Step 3:[x^q の更新]

参照点 x^p を除いた探索点群 X から更新点 x^q を選び、更新点の評価値と近傍解の評価値と比較し、更新点の評価値が優れている場合に更新する。

$$x^q := \begin{cases} \hat{x} & f(\hat{x}) \leq f(x^q) \\ x^q & \text{otherwise} \end{cases}$$

Step 4:[排斥]

排斥確率 p_a に従い、最悪点 $x^w = \arg \max_{x \in X} \{f(x)\}$ を排斥させる。

$$x^w := x^w + \alpha L(\beta)$$

$$L(\beta) = [L_1(\beta), L_2(\beta), \dots, L_n(\beta)]^T$$

最悪点の評価値 $f(x^w)$ を求め、 $k := k + 1$ とする。

Step 5:[終了判定]

$k \geq k_{\max}$ の時に、探索を終了する。もしなければ、**Step 2** へ戻る。

一般的なメタヒューリスティクスの探索機構は主に「近傍生成」と「解の更新」がある。

CS の近傍解の生成はランダムに選択した参照点 x^p がレヴィ乱数を通して近傍解 \hat{x} を生成する。なお、解の更新では更新点 x^q が参照点 x^p を抜いた探索点群から選択し、更新点の評価値と近傍解 \hat{x} の評価値と比較して近傍解が優れている場合に更新する。一般的なメタヒューリスティクスでは参照点 x^p と更新点 x^q は同一の点であるが、CS の参照点 x^p と更新点 x^q は異なっている。

また、一般的なメタヒューリスティクスにおいて、更新点の情報により近傍解を生成するが、CS はほかの参照点の情報だけで近傍解を生成する。

2.2 探索戦略

2.2.1 近接最適性原理

近接最適性原理（Proximate Optimality Principle : POP）とは「良い解同士は類似構造を持つ」という原理である[10]。こちらでは近い距離の解同士は類似構造があると考えている。良い解の付近に良い解の存在する可能性が高い。CS の探索の中にもし参照点は良い解を持っていると、POP によってこの参照点の付近に良い解の存在する可能性が高い、参照点によって近傍生成範囲を縮小して、集中化する。もし参照点は悪い解を持っていると、POP によって参照点の付近に良い解の存在する可能性が低い、参照点によって近傍生成範囲を拡大して、多様化する。

- 「良い解」＝優れた評価値をもつ解
- 「類似構造」＝解同士の距離が近いこと

と解釈する。

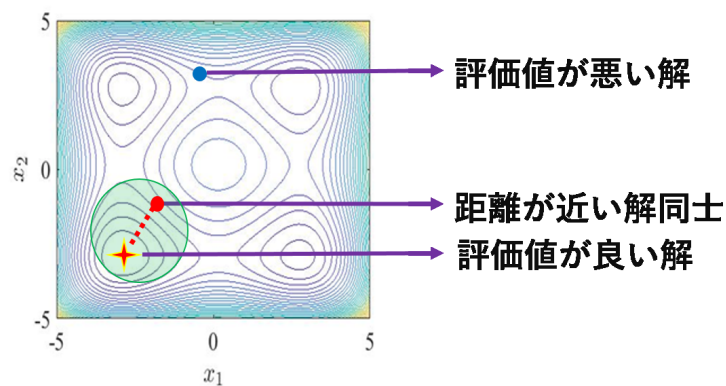


図 2.1 : 近接最適性原理の概図

2.2.2 多様化・集中化

集中化は解空間の有望領域を重点的に探索する，短期的な解の改善を目的とする探索方針である。多様化は，解区間の広い範囲を探索する，長期的な解の改善を目的とする探索方針である。探索時間から見ると，探索初期は探索の多様化を実現，探索終盤は探索の集中化を実現する。探索空間から見ると，よい探索空間に入った時に集中探索して集中化を実現，悪い探索空間に入った時により広い範囲で探索して多様化を実現する。

- 空間的な多様化：解区間の広い範囲を探索，長期的な解を改善。
- 空間的な集中化：解区間の狭い範囲を探索，短期的な解を改善。

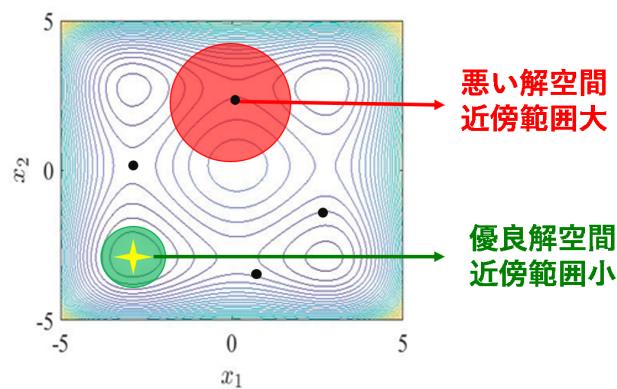


図 2.2：空間的な多様化・集中化

- 時間的な多様化：探索序盤で探索点をより広い範囲で探索，多様化を実現して長期的な解の改善。
- 時間的な集中化：探索終盤で探索点を良い点の付近に集中して探索，集中化を実現して短期的な解の改善。

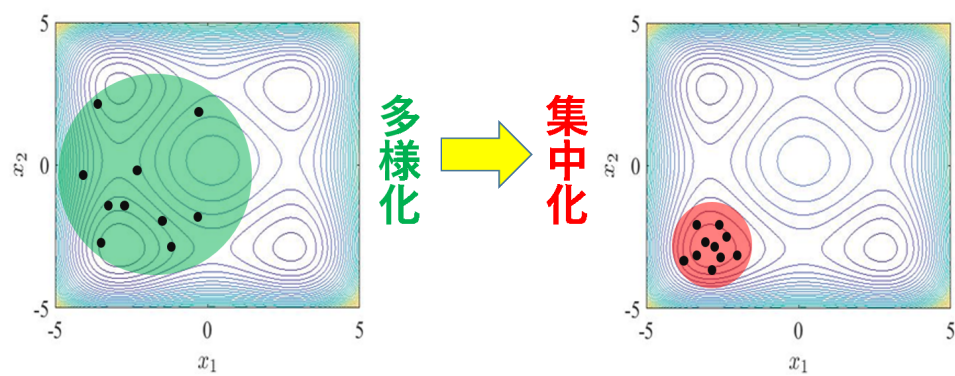


図 2.3 : 時間的な多様化・集中化

3

Cuckoo Search における 先行研究と課題

本章では Cuckoo Search の適応性を向上する先行研究の位置付けを述べる。Cuckoo Search の探索に関わるパラメータを解析し、Cuckoo Search の特徴によって妥当な探索戦略で CS の適応化を実現する。最後に先行研究の課題について述べる。

3.1 先行研究の位置付け

代表的なメタヒューリスティクスの Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) [1, 3]などは、ほとんど複数の探索点で探索を行っている。そして、探索する時にすべての探索点は同じなパラメータで探索している[1, 7, 10]。もし一様でないパラメータを用いることで効率的な探索が実現出来れば、アルゴリズムの適応性と性能を向上できると考えられる。しかし、パラメータの自由度を高めるために、適切なパラメータ調整則が必要である。

先行研究では、上述の多点探索型メタヒューリスティクスにおけるパラメータ設定の自由度の拡張と、適切なパラメータの調整機能の付加の両立の重要性を踏まえ、多点探索型メタヒューリスティクスの一手法である Cuckoo Search (CS) [8, 9]の探索性能向上を実現した。具体的には、CS における各探索点のパラメータを多様化するとともに、各探索点のランキングに基づく適応的なパラメータ調整機能を付加した新たな適応型 CS を提案した。

3.2 Cuckoo Search のパラメータ解析と調整

3.2.1 Cuckoo Search のパラメータ解析

CS の探索に影響できるパラメータは主に二つのパラメータ α と β がある。パラメータ α と β で CS の探索を制御することができる。先行研究では、CS のパラメータの自由度と適応性を実現するために、まず一般的なメタヒューリスティクス手法の観点から CS の探索構造（近傍解の生成、解の更新）とパラメータの影響を解析した。そして、CS アルゴリズムに対して解析した後に、結果によって CS のパラメータの自由度と適応性がまだ低いと考えられる。そこで、先行研究においては主に CS アルゴリズムのパラメータの自由度と適応性の向上の改良を行った[12, 13, 14, 15]。

CS の探索の中に CS アルゴリズムのステップ幅のパラメータ α はアルゴリズムの探索の摂動スケールに影響できる。 α が大きいほど、探索摂動スケールも大きくなる。また、もう一つのパラメータ β は CS アルゴリズムの近傍生成範囲の大きさに影響できる。パラメータの調整することで近傍解は探索空間の良さの情報によって解空間の任意位置での生成確率を変えることができる。

ここでは、先行研究では主に調整しているパラメータ β について解析を行う。パラメータ β の値と近傍解の生成範囲の変化の関係を示す。

図 4.1 は CS のパラメータ β の値に応じた近似レヴィ分布図である。近似レヴィ分布図によりパラメータ β の値によって近似レヴィ分布の確率的に値の生成範囲を変化することができる。示した図から見ると、パラメータ β の値が大きい場合に、近似レヴィ分布 $L(\beta)$ の値は小さい値を取る可能性が高い、大きい値を取る可能性が低い。

一方、パラメータ β の値が小さい場合に、近似レヴィ分布 $L(\beta)$ の値は大きい値を取る可

能性が高い、小さい値を取る可能性が低い。そこで、 β が大きい場合に、参照点 x^p によって近似レヴィ分布は小さい値を取る可能性が高い、近傍解 \hat{x} の生成範囲が小さくなり、 β が小さい場合に、参照点 x^p によって近似レヴィ分布は大きい値を取る可能性が高い、近傍解 \hat{x} の生成範囲が大きくなる。

以上から、CS アルゴリズムの探索の中に β の変化により、近似レヴィ分布の値を取る確率が変わり、パラメータ β の変化で近傍解 \hat{x} の生成範囲の大きさも変わることが理解できる。

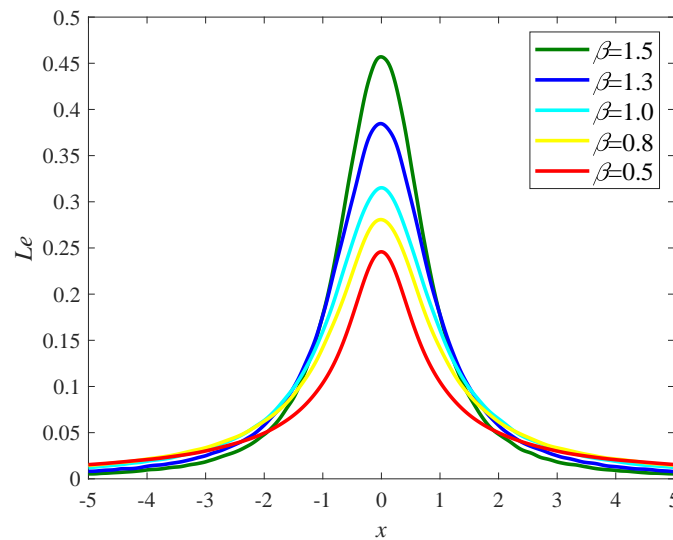


図 3.1：近似レヴィ分布

3.2.2 Cuckoo Search のパラメータ調整

CS のパラメータを解析する結果から見ると、CS の近傍生成範囲はパラメータ α と β によって調整できる。従来の CS は探索する時に、全ての探索点のパラメータが同じであり、探索空間の良さの情報に依り調整がなくて、探索の効率が低い。もし探索点の探索情報によって一様ではないパラメータをつけたら、パラメータの適応性を実現できる。そして、CS アルゴリズムの探索性能を向上することができると思う。

先行研究では、CS のパラメータの適応的に調整するために、近接最適性原理 (Proximate

Optimality Principle : POP) と多様化・集中化の戦略を使って、具体的に以下の調整則を提案した。

- 探索点の評価値が悪い場合に、近傍解の生成範囲を大きくして多様化する
- 探索点の評価値が良い場合に、近傍解の生成範囲を小さくして集中化する

上述の調整は評価値の情報により、近傍生成範囲を調整する。近接最適性原理 POP と空間的な多様化・集中化戦略を結合することを通して、CS のアルゴリズムの探索性能の向上が期待できる。

3.3 探索点のランキングに基づく先行研究の提案

以上の探索戦略から、先行研究では各探索点の評価値の良さによって探索点にランキングをつけてパラメータが適応的に調整できる Cuckoo Search を提案した。探索点が現在持っている解情報によって、 β を適応的に調整して近傍解の生成範囲を自動的に調整する。具体的な β の調整則を式 (3.1) に示す。

$$\beta^i := \beta_{\max} - (\beta_{\max} - \beta_{\min}) \frac{R^i - 1}{m - 1} \quad (i = 1, 2, \dots, m) \quad (3.1)$$

ここで、 R^i は探索点 x^i の評価値の順位、 m は探索点数、 β_{\max} および β_{\min} はそれぞれ β の最大値、最小値を表す。オリジナルの CS では、全ての探索点が一様のパラメータ β で探索を行っている。つまり探索中にアルゴリズムの近傍解の生成範囲もずっと同じとなっている。探索点の探索情報によって探索することができない。オリジナル CS の探索効率が低いとなる。

それに対して、先行研究の CS ではパラメータの自由度を高め、探索点の探索情報により適応的にパラメータ設定できていて、近接最適性原理 (POP) と結合した空間的な多様化・集中化が実現している。

空間的な多様化・集中化の CS の近傍解の生成範囲を図 5.1 に示す。先行研究の CS においては、優れた評価値を持つ探索点の近傍範囲は小さくして集中化する。良くない評価値を持つ探索点の近傍範囲は大きくして多様化する。先行研究の CS のパラメータ β が探索

情報により，適応的に調整されている。

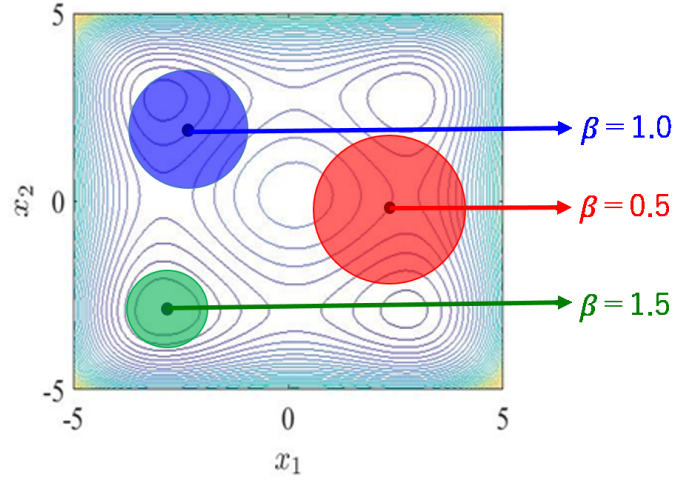


図 3.2：近傍解の生成範囲の概略図

以下に，先行研究で提案する適応型 CS のアルゴリズムを示す[13, 14]。

【適応型 Cuckoo Search】

Step 0:[準備]

探索点数 m ，ステップサイズ $\alpha > 0$ ，排斥率 $p_a \in [0, 1]$ ，最大評価回数 k_{\max} ， β_{\min} ， β_{\max} を決め，評価回数 $k = 0$ とする。

Step 1:[初期化]

探索点の初期解 $\mathbf{x}^i (i = 1, 2, \dots, m)$ を初期配置領域 S からランダムに生成し，探索点群を $\mathbf{X} = \{\mathbf{x}^i | i = 1, 2, \dots, m\}$ とする。探索点の評価値 $f(\mathbf{x}^i)$ を計算し， $k := m$ とする。

Step 2:[レヴィフライト]

探索点 \mathbf{x}^i のパラメータ β^i を以下の式で設定する。

$$\beta^i := \beta_{\max} - (\beta_{\max} - \beta_{\min}) \frac{R^i - 1}{m - 1}$$

探索点群からランダムに一つの参照点 $\mathbf{x}^p \in \mathbf{X}$ を選択し，以下の式により近傍解 $\hat{\mathbf{x}}$

を生成する。

$$\hat{\mathbf{x}} := \mathbf{x}^p + \alpha \mathbf{L}(\beta^i)$$

$$\mathbf{L}(\beta^i) = [L_1(\beta^i), L_2(\beta^i), \dots, L_n(\beta^i)]^T$$

近傍解の評価値 $f(\hat{\mathbf{x}})$ を計算し, $k := k + 1$ とする。

Step 3:[更新]

参照点を除いた探索点群から更新点 $\mathbf{x}^q (\neq \mathbf{x}^p)$ を選び, そして, 以下の式により更新を行う。

$$\mathbf{x}^q := \begin{cases} \hat{\mathbf{x}} & f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^q) \\ \mathbf{x}^q & \text{otherwise} \end{cases}$$

Step 4:[排斥]

排斥率 p_a に従い, 以下の式で最悪点 $\mathbf{x}^w = \arg \max_{\mathbf{x} \in X} \{f(\mathbf{x})\}$ を排斥させる。

$$\mathbf{x}^w := \mathbf{x}^w + \alpha \mathbf{L}(\beta_{\max})$$

$$\mathbf{L}(\beta_{\max}) = [L_1(\beta_{\max}), L_2(\beta_{\max}), \dots, L_n(\beta_{\max})]^T$$

最悪点の評価値 $f(\mathbf{x}^w)$ を計算し, $k := k + 1$ とする。

Step 5:[終了判定]

$k \geq k_{\max}$ であれば, 探索が終了する。もしなければ **Step 2** へ戻る。

3.4 先行研究の中に残った課題

一般的にメタヒューリスティクスの探索は時間的な観点から探索状況が以下のことになることが考えられる。

- 探索序盤：探索が不十分な状況で, 良い評価値を持つ探索点が少ない可能性が高い。
- 探索終盤：探索が十分な状況で, 良い評価値を持つ探索点が多い可能性が高い。

先行研究が提案した適応型 CS では探索点の探索状況を考慮した POP 戦略に基づいて空間的な多様化・集中化を実現したが, 時間軸から見ると, 図 3.3 のようにこの空間的な多様

化・集中化に基づく制御が探索過程を通して同じである。もし図 3.4 のように探索のイテレーションに応じて制御が同じではないなら，アルゴリズムの探索性能を向上することが考えられる。イテレーションに応じて制御が一様である探索から，先行研究の適応型 CS は以下のような問題が生じる。

- 探索序盤：悪い評価値を持つ探索点がランキングの上位に存在する可能性が高いにも関わらず，それらの探索点の近傍解の生成範囲が狭い。
- 探索終盤：良い評価値を持つ探索点がランキングの下位に存在する可能性が高いにも関わらず，それらの探索点の近傍解の生成範囲が広い。

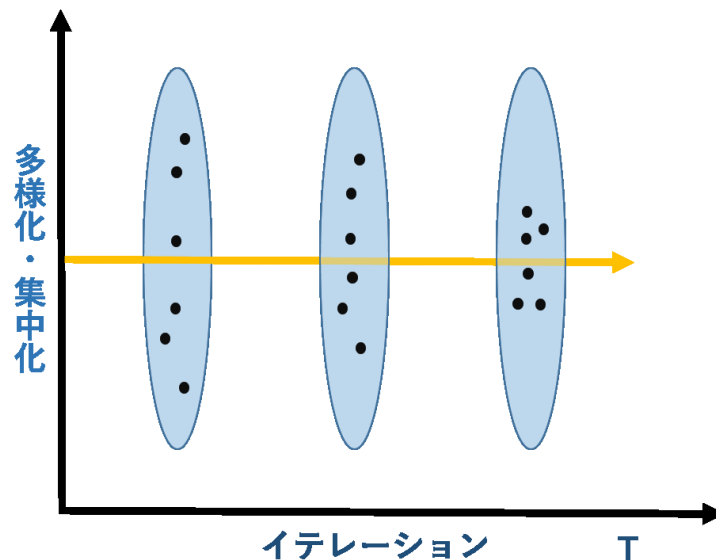


図 3.3：イテレーションに応じて制御が一様である探索

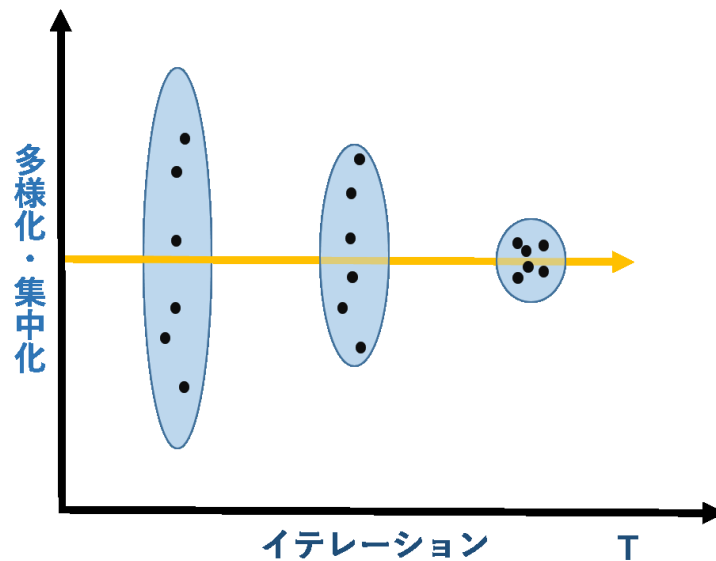


図 3.4：イテレーションに応じて制御が一様ではない探索

先行研究において提案された「空間的な多様化・集中化に基づく適応型 Cuckoo Search」でも、ある程度の時間的な多様化・集中化を実現できているが、先行研究の多様化・集中化は、初期値の配置、ベンチマーク関数の種類、探索点数などからの影響があり、常に十分な多様化・集中化を実現することはできない。先行研究では、探索終了する時に、集中化できないことがあるし、探索のイテレーション回数の半分ぐらいで強く集中化して、多様化が不足することもある。図 3.3 のように、先行研究の空間的な多様化・集中化では探索終了する時に、緑の線のように集中化できないことがある一方で、赤の線のように探索のイテレーション回数の半分位で強く集中化することもある。

以上から、先行研究が提案した適応型 CS は十分な時間的な多様化・集中化が実現できていない。

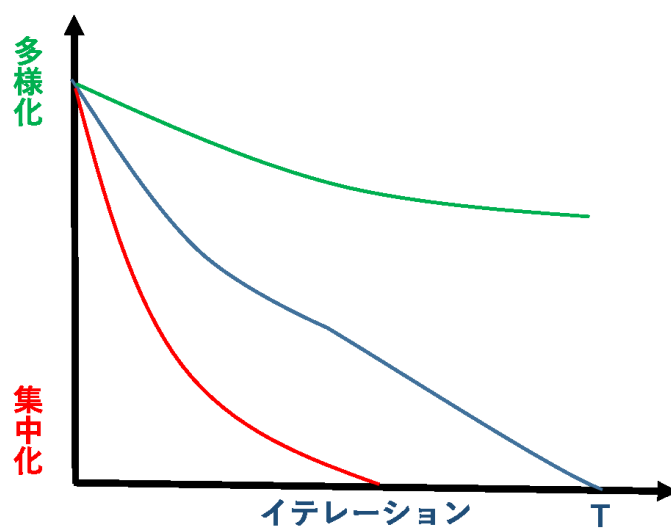


図 3.5 : 先行研究の多様化と集中化の程度

4

時空間の多様化・集中化 に基づく適応型 Cuckoo Search

本章では，本研究の位置付けを述べる。また，CS の近傍生成と排斥の時空間の多様化・集中化の実現について説明する。十分な時空間の多様化・集中化を有する CS を提案し，数値実験により提案手法の性能向上を検証する。

4.1 本研究の位置付け

空間的な多様化・集中化機能を有する先行研究では，CS の空間軸の観点から見ると適応性が向上したが，時間軸の観点から見ると，多様化・集中化に基づく制御が同じ，つまり毎回の探索においてパラメータ調整則が同じになっている。そのため，反復回数の変化に伴い，探索状態の制御が十分に行うことができないという課題を残している。先行研究においても，ある程度の時間的な多様化・集中化を実現できるが，その多様化・集中化は不十分である。先行研究では，探索終了する時に，集中化できないことがあるし，多様化が足りないこともある。先行研究の多様化・集中化は，探索点の初期値，問題の構造，探索点数などの影響があり，十分な多様化・集中化を実現することはできないため，時間的な多様化・集中化のレベルが低い。具体的には，時間軸の観点から以下のような戦略が必要である。

- 探索序盤：探索点全体の傾向として，近傍解の生成範囲が広い
- 探索終盤：探索点全体の傾向として，近傍解の生成範囲が狭い

本研究では上述の視点に立ち、これらの課題を解決するためにCSの新たなパラメータ調整戦略として、先行研究の成果を踏まえ、近傍生成に対してレベルが高い時間的な多様化・集中化機能と新たな近傍生成方法を提案する。さらに、CSの排斥の操作には先行研究の探索戦略と本研究の探索戦略を同時に活かすことを可能とする新たな排斥の調整則も提案する。

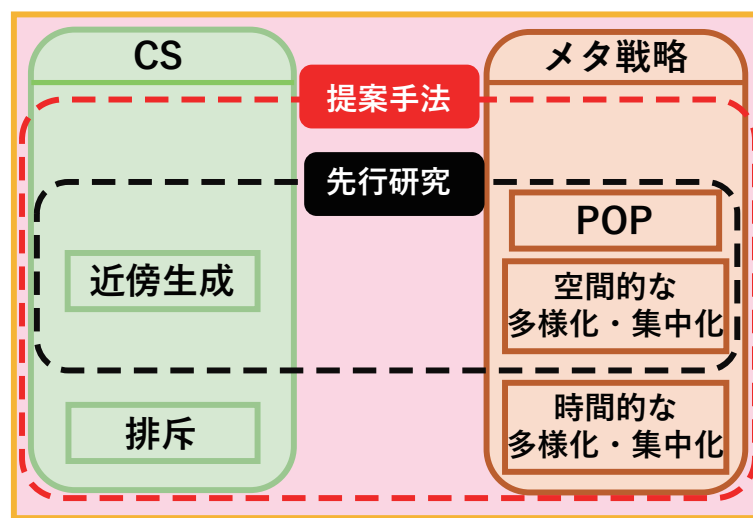


図 4.1：研究の位置付け

本研究で提案する新たなCSは、時間的な多様化・集中化戦略として、探索序盤ではより広い範囲で探索することで多様化を実現して長期的な解の改善を実現する一方で、探索終盤では良い解付近を集中的に探索することで短期的な解の改善を実現する。具体的には、空間的な多様化・集中化を実現するために、各探索点の評価値に応じてランキングに基づいてパラメータを調整する戦略と、時間的（反復回数に応じた）な多様化・集中化戦略を有機的に結合した新たな適応型CSを提案する。本研究では探索のイテレーション回数を問わずに、探索初期の十分の多様化と探索終盤の十分の集中化は両方保証できる。本研究はベンチマーク関数の種類、探索の次元数、初期値などを問わずに、高いレベルで多様化・集中化を確実に実現できる。そして、典型的なベンチマーク問題を用いた数値実験により、オリジナルCSおよび先行研究のCSと比較しつつ、提案手法の探索性能を詳細に検証・評価する。

4.2 CSの近傍生成における時空間の多様化・集中化の実現

先行研究ではCuckoo Searchの近傍生成に対して探索点のランキングに基づくパラメータを適応的に調整している。近傍解の生成範囲の調整を行うパラメータとして β を用い、探索点のランキングを基に β を調整する。先行研究の β の調整則を式(4.1)に示す。

$$\beta^i := \beta_{\max} - (\beta_{\max} - \beta_{\min}) \frac{R^i - 1}{m - 1} \quad (i = 1, 2, \dots, m) \quad (4.1)$$

ここで、 R^i は探索点 x^i の評価値の順位、 m は探索点数、 β_{\max} および β_{\min} はそれぞれ β の最大値、最小値を表す。この式により、先行研究では探索点の探索状況によってパラメータ β を変化させて近傍生成範囲を変化させることができる。しかし、アルゴリズムの探索に対してパラメータ調整則が同じになっている。したがって、時間的に探索状態の制御が十分に行えないという課題が残している。時間的な多様化・集中化が不十分である。一般的なメタヒューリスティクス手法では、探索序盤に多くの探索点の評価値が良くなく、この時に多様化を実現してより広い範囲で探索すべきである。探索終盤に多くの探索点の評価値が優れており、この時に集中化を実現して良い探索点の付近で探索すべきである。

先行研究の空間的な多様化・集中化とパラメータ多様性の機能を保持するために、ランキングに基づいてパラメータ β を調整する上で探索時間のとともに、パラメータ β の下限を調整して、パラメータ β の変化範囲を調整する。具体的な β の調整則を式(4.2)、式(4.3)に示す。

$$\beta_{\min 2} = \beta_{\min} + 0.8 \times (\beta_{\max} - \beta_{\min}) \frac{k}{k_{\max}} \quad (4.2)$$

$$\beta^i := \beta_{\max} - (\beta_{\max} - \beta_{\min 2}) \frac{R^i - 1}{m - 1} \quad (i = 1, 2, \dots, m) \quad (4.3)$$

ここで、 k は探索の評価回数、 k_{\max} は最大評価回数である。この式により、先行研究の空間的な多様化・集中化の機能を維持すると同時に、評価回数の増加とともに、十分に探索の時間的な多様化から集中化への調整が実現できる。

4.3 CSの排斥における時空間の多様化・集中化の実現

先行研究では排斥率をゼロとしたが、実際はCSの排斥操作は探索の多様性を強めることができる。排斥に対して適当な調整を行うことが可能であれば、よりCSの探索性能の向上が期待できる。排斥操作に対して、まず排斥点の選点方式について改良を行った。従来の排斥点の選点方式では探索点の中で評価値が一番悪いものを選択して排斥する。評価値が一番悪い探索点だけ排斥を行うのであれば、探索点の探索情報によって、排斥の空間的な多様化・集中化を実現することが難しい。排斥操作の空間的な多様化・集中化を実現するために、今回は探索点の適応度比例 e^i で排斥点をルーレット選択する。評価値情報が良い探索点は適応度が高く、排斥される可能性が小さい。評価値情報が悪い探索点は適応度が低く、排斥される可能性が大きい。ランキングが R の探索点の適応度比例は式(4.4)で計算する。

$$e^R = \frac{f(x^R) - f(x)_{best}}{\sum_{j=2}^m \{f(x^j) - f(x)_{best}\}} \quad (R = 2, 3, \dots, m) \quad (4.4)$$

ランキングが R の点のルーレット選択の上限は式(4.5)で計算する。

$$S_R = \sum_{j=2}^R e^j \quad (i = 2, 3, \dots, m) \quad (4.5)$$

(0,1) からランダムに数値 r を生成する。もし

$$S_R < r \leq S_R \quad (4.6)$$

なら、 x^R を排斥する。

具体的な排斥する探索点を選択するステップは：

- (1) m 個の探索点の評価値が一番良い探索点の評価値 f_1 から一番悪い探索点の評価値 f_m まで $(f_1 \dots f_m)$ を計算する。
- (2) 評価値が一番良い探索点の評価値 f_1 と他の $m-1$ 個の点の評価値の差を計算する $(k_2 = f_2 - f_1 \dots k_m = f_m - f_1)$ 。
- (3) すべての評価値差の和 T を計算する。 $(T = k_2 + k_3 + \dots k_m)$
- (4) $R_1 = k_2, R_2 = k_2 + k_3, R_3 = k_2 + k_3 + k_4, \dots, R_{m-1} = k_2 + k_3 + k_4 + k_5 + \dots + k_m$ を計算する

- (5) $S_1 = R_1/T, S_2 = R_2/T, \dots, S_{m-1} = R_{m-1}/T$ の探索点の適応度を計算する。
- (6) 評価値が一番良い探索点を選択せず保存する。 $(0, 1)$ からランダム r を生成する。もし r の値が $(0, S_1)$ の範囲内は評価値が2番目良い点を選択する。もし r の値が (S_{m-2}, S_{m-1}) の範囲内ならば評価値が一番悪い探索点を選択する。この方法では、評価値が悪い探索点を選択される確率が高い。このように、探索点の良さによって探索点を選択して排斥することができる。

そして、選択した排斥点に対して近傍生成と同じ探索戦略を使って、CSの排斥操作の時空間の多様化・集中化を実現する。選択した排斥点に対しての具体的な β の調整則を式(4.7), 式(4.8)に示す。

$$\beta_{\min 3} = \beta_{\min} + 0.8 \times (\beta_{\max} - \beta_{\min}) \frac{k}{k_{\max}} \quad (4.7)$$

$$\beta^{pa} := \beta_{\max} - (\beta_{\max} - \beta_{\min 3}) \frac{R^{pa} - 1}{m - 1} \quad (4.8)$$

ここで、 k は探索の評価回数、 β^{pa} は排斥点に対してのパラメータ β 、 R^{pa} は排斥点 \mathbf{x}^{pa} の評価値の順位、 k_{\max} は最大評価回数である。以上の調整方法で排斥操作における時空間の多様化・集中化を実現することができると考えられる。

以下に、本研究で提案する時空間の多様化・集中化に基づく適応型Cuckoo Searchのアルゴリズムを示す。

【時空間の多様化・集中化に基づく適応型Cuckoo Search】

Step 0:[準備]

探索点数 m , ステップサイズ調整パラメータ $\alpha > 0$, 排斥確率 $p_a \in [0, 1]$, 最大評価回数 k_{\max} , β_{\min} , β_{\max} を定め、評価回数 $k = 0$ とする。

Step 1:[初期化]

探索点の初期解 $\mathbf{x}^i (i = 1, 2, \dots, m)$ を初期配置領域 S 内にランダムに与え、探索点群を $\mathbf{X} = \{\mathbf{x}^i | i = 1, 2, \dots, m\}$ とする。探索点の評価値 $f(\mathbf{x}^i)$ を求め、 $k := m$ とする。

Step 2:[レヴィフライト]

探索点 \mathbf{x}^i のパラメータ β^i を以下の式で設定する。

$$\beta_{\min 2} = \beta_{\min} + 0.8 \times (\beta_{\max} - \beta_{\min}) \frac{k}{k_{\max}}$$

$$\beta^i := \beta_{\max} - (\beta_{\max} - \beta_{\min 2}) \frac{R^i - 1}{m - 1} \quad (i = 1, 2, \dots, m)$$

探索点群から参照点 $\mathbf{x}^p \in \mathbf{X}$ をランダムに 1 つ選択し、以下の式で近傍解 $\hat{\mathbf{x}}$ を生成する。

$$\hat{\mathbf{x}} := \mathbf{x}^p + \alpha \mathbf{L}(\beta^i)$$

$$\mathbf{L}(\beta^i) = [L_1(\beta^i), L_2(\beta^i), \dots, L_n(\beta^i)]^T$$

近傍解の評価値 $f(\hat{\mathbf{x}})$ を求め、 $k := k + 1$ とする。

Step 3:[更新]

参照点を除いた探索点群から更新点 $\mathbf{x}^q (\neq \mathbf{x}^p)$ を選び、以下の式により更新する。

$$\mathbf{x}^q := \begin{cases} \hat{\mathbf{x}} & f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^q) \\ \mathbf{x}^q & \text{otherwise} \end{cases}$$

Step 4:[排斥]

排斥確率 p_a に従い、探索点の適応度比例 e^i で排斥点をルーレット選択する。ランキングが R の探索点の適応度比例は以下の式で計算する。

$$e^R = \frac{f(x^R) - f(x)_{best}}{\sum_{j=2}^m \{f(x^j) - f(x)_{best}\}} \quad (R = 2, 3, \dots, m)$$

選択した排斥点 \mathbf{x}^{pa} は以下の操作で移動させる。

$$\beta_{\min 3} = \beta_{\min} + 0.8 \times (\beta_{\max} - \beta_{\min}) \frac{k}{k_{\max}}$$

$$\beta^{pa} := \beta_{\max} - (\beta_{\max} - \beta_{\min 3}) \frac{R^{pa} - 1}{m - 1}$$

$$\mathbf{x}^{pa} := \mathbf{x}^{pa} + \alpha \mathbf{L}(\beta^{pa})$$

$$\mathbf{L}(\beta^{pa}) = [L_1(\beta^{pa}), L_2(\beta^{pa}), \dots, L_n(\beta^{pa})]^T$$

排斥点の評価値 $f(\mathbf{x}^{pa})$ を求め、 $k := k + 1$ とする。

Step 5:[終了判定]

$k \geq k_{\max}$ ならば、探索を終了する。さもないければ **Step 2** へ戻る。

4.4 数値実験

4.4.1 数値実験条件

6種類の典型的なベンチマーク関数を用いた数値実験を行い、オリジナルCSと先行研究の空間的な多様化・集中化を有するCSと比較することで、提案手法の探索性能を検証する。用いたベンチマーク関数 2ⁿ minima, Rastrigin, Levy, Ackley, Griewank, Alpine については付録に示す。今回選択したベンチマーク関数は主に代表的な多峰関数である。多峰関数は提案手法の大域的な最適解の探索性能を検証することができる。

共通条件として、探索点数 $m=10$, 次元数 $N=50, 100, 300, 1000$ とする。ステップサイズ調整変数 $\alpha=0.1$, 最大評価回数 $k_{max}=5000$ とする。また、オリジナルCSでは $\beta=1.5$ (推薦値) を設定する。先行研究と提案手法は下限値 $\beta_{min}=0.3$, 上限値 $\beta_{max}=1.99$ を用いる。50 試行回数で探索過程における最良の評価値 $f(\mathbf{x})$ の平均値 (Mean), 最良値 (Best), 最悪値 (Worst), 標準偏差 (S.D.) をそれぞれ示す。また、排斥率 Pa については、オリジナルCS, 先行研究CSの排斥率 $Pa = 0$ とし、提案手法の排斥率 $Pa = 1$ とする。

4.4.2 実験結果・考察

実験結果の図 4.1-図 4.6 と表 4.1-表 4.6(CS:オリジナルCS, PR:先行研究のCS, ACS1:提案手法 $Pa = 0$ のCS, ACS2:提案手法 $Pa = 1$ のCS)により、排斥率を1とした提案手法はオリジナルCS, 先行研究のCS, 排斥率を0とした提案手法より多くの結果が優れている。特に1000次元の時に提案手法の結果はほとんど優れている。実験結果から、先行研究の時間的な多様化・集中化が十分ではない課題を解決したことがわかる。さらに、新たな排斥手法の性能も検証した。システムの大規模化・複雑化の発展の中、提案手法は高次元で優れた探索性能を発揮できると考えられる。そして、本研究における提案手法ではパラメータの自由度と多様化を実現している。ユーザーはアルゴリズムを使う際に、パラメータの設定の考える必要がない。つまりアルゴリズムの使用の利便性も向上した。

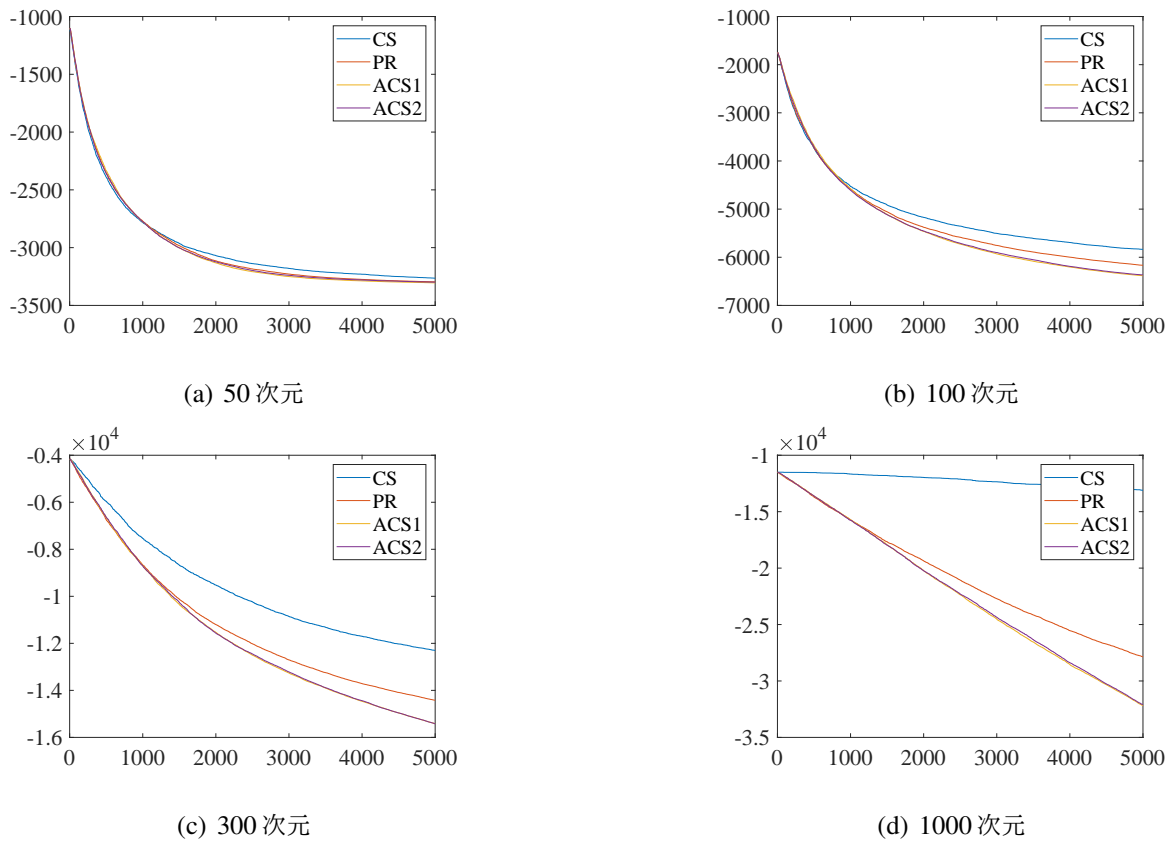


図 4.2 : 2^n minima におけるそれぞれの手法に対して平均値の推移図 (オリジナル CS : 青, 先行研究 : 赤, 提案手法 ($Pa=0$) : オレンジ, 提案手法 ($Pa=1$) : 紫)

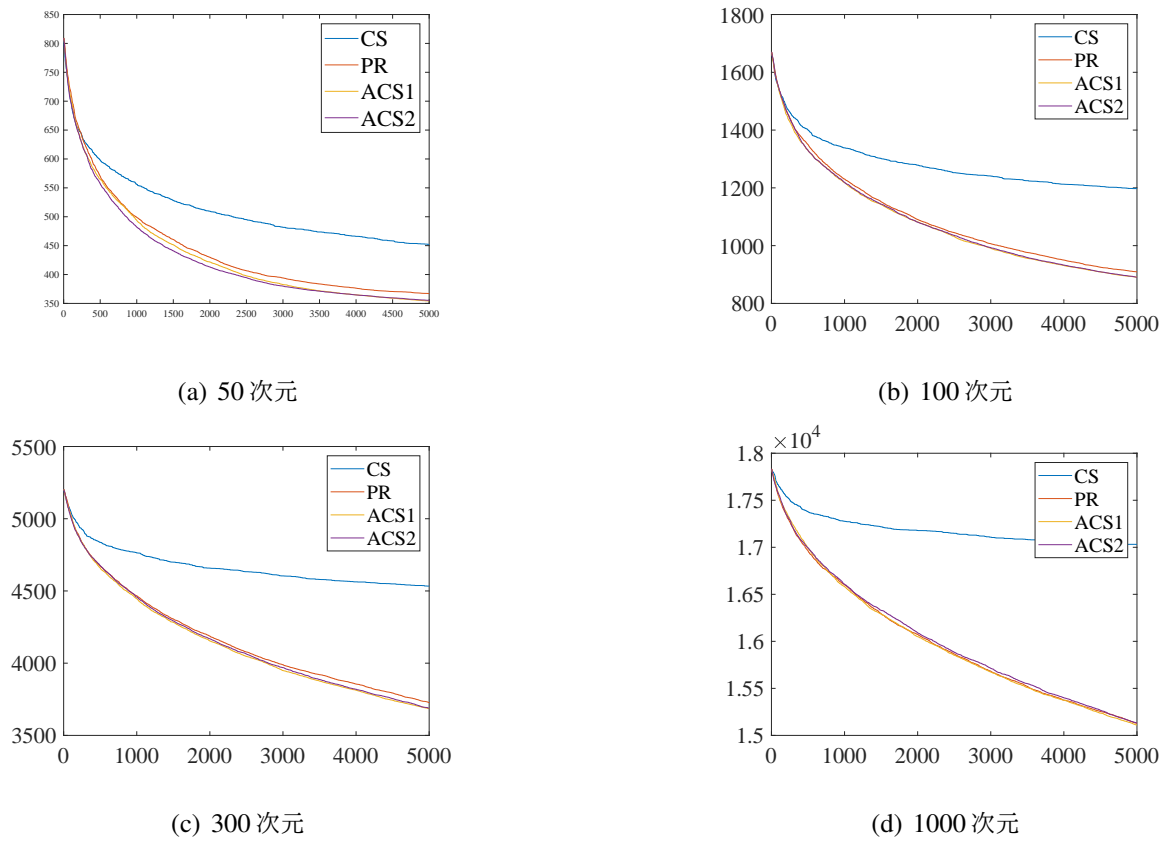
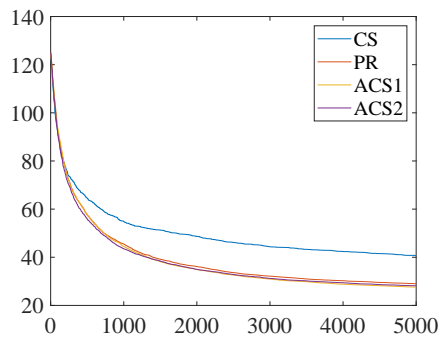
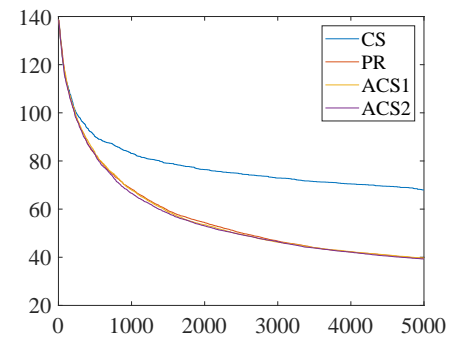


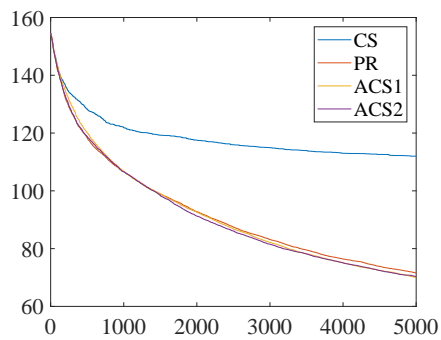
図 4.3 : Rastrigin におけるそれぞれの手法に対して平均値の推移図 (オリジナル CS : 青, 先行研究 : 赤, 提案手法 ($Pa=0$) : オレンジ, 提案手法 ($Pa=1$) : 紫)



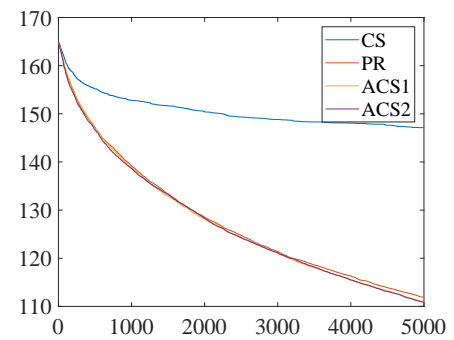
(a) 50次元



(b) 100次元



(c) 300次元



(d) 1000次元

図4.4 : Levyにおけるそれぞれの手法に対して平均値の推移図（オリジナルCS：青，先行研究：赤，提案手法 ($Pa=0$)：オレンジ，提案手法 ($Pa=1$)：紫）

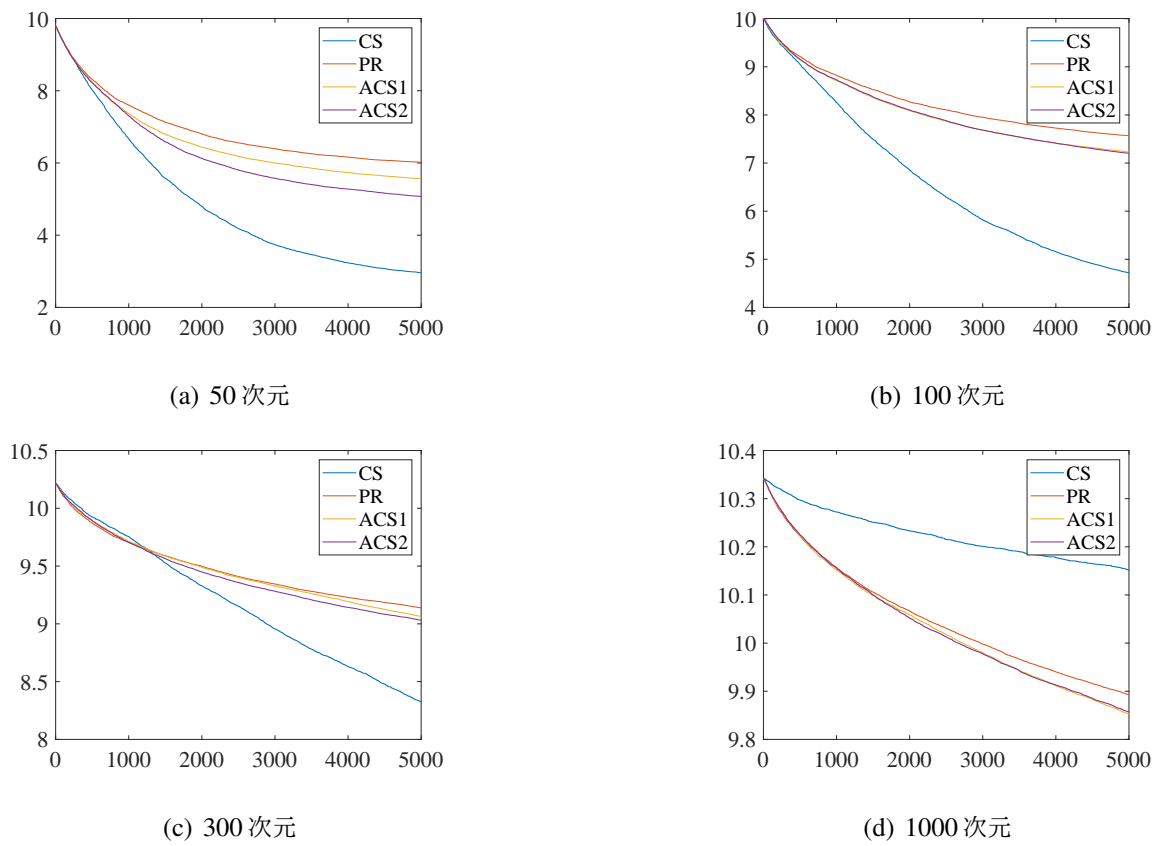
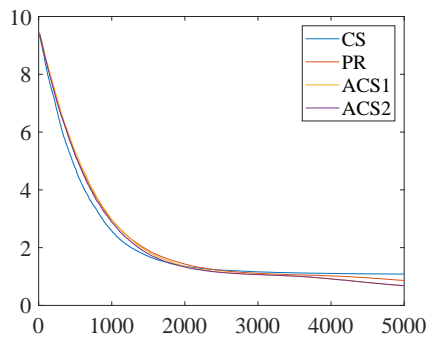
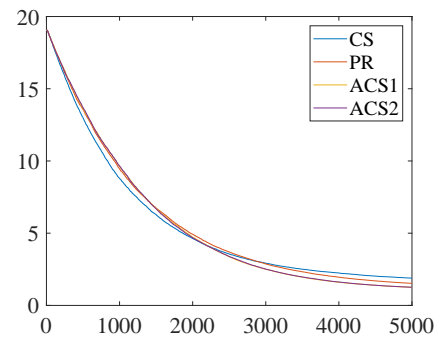


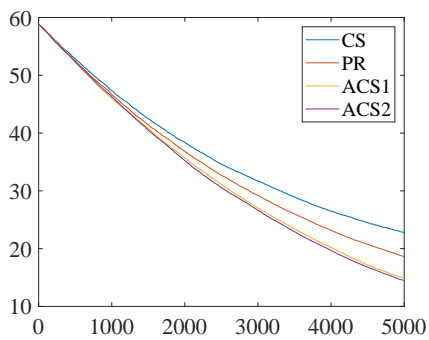
図 4.5 : Ackley におけるそれぞれの手法に対して平均値の推移図 (オリジナル CS : 青, 先行研究 : 赤, 提案手法 ($Pa=0$) : オレンジ, 提案手法 ($Pa=1$) : 紫)



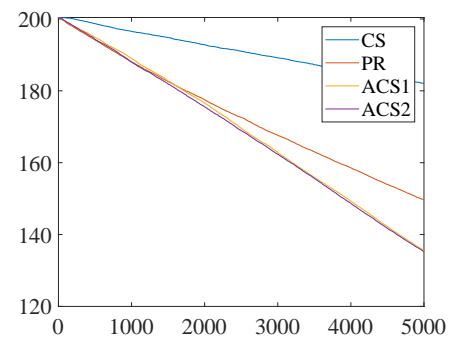
(a) 50 次元



(b) 100 次元

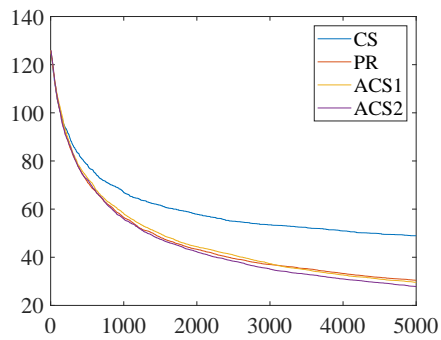


(c) 300 次元

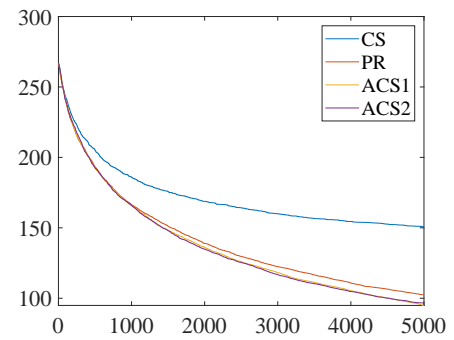


(d) 1000 次元

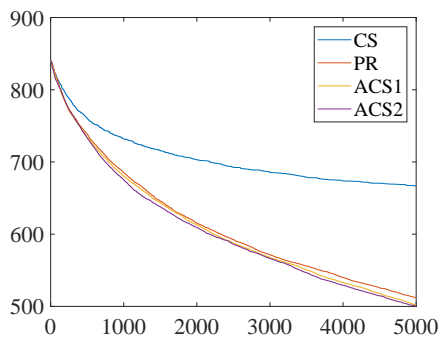
図 4.6 : Griewank におけるそれぞれの手法に対して平均値の推移図 (オリジナル CS : 青, 先行研究 : 赤, 提案手法 ($Pa=0$) : オレンジ, 提案手法 ($Pa=1$) : 紫)



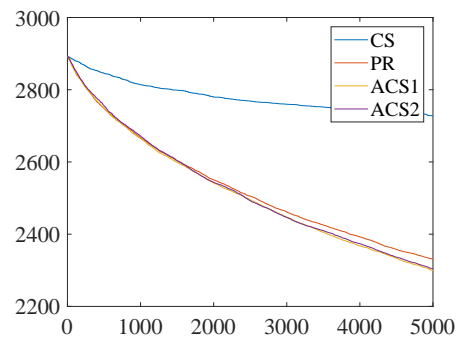
(a) 50 次元



(b) 100 次元



(c) 300 次元



(d) 1000 次元

図 4.7 : Alpine におけるそれぞれの手法に対して平均値の推移図 (オリジナル CS : 青, 先行研究 : 赤, 提案手法 ($Pa=0$) : オレンジ, 提案手法 ($Pa=1$) : 紫)

表 4.1 : 2^n minima おける数値実験の結果

	次元数	手法	平均値	最良値	最悪値	標準偏差
2^n minima	50	CS	-3264	-3450	-3056	975
		PR	-3308	-3535	-3072	100.1
		ACS1	-3304	-3593	-3084	120.7
		ACS2	-3320	-3537	-3059	98
	100	CS	-5837	-6065	-5566	114.3
		PR	-6167	-6495	-5885	146.8
		ACS1	-6383	-6649	-5970	147.7
		ACS2	-6358	-6684	-6039	141
	300	CS	-12188	-12924	-11417	393
		PR	-14419	-15077	-13668	311
		ACS1	-15418	-16145	-14798	309
		ACS2	-15431	-16292	-14758	336
	1000	CS	-13107	-16431	-10732	1179
		PR	-27873	-30970	-25375	2537
		ACS1	-32192	-34839	-29799	1283
		ACS2	-32290	-34941	-30164	1071

表 4.2 : Rastrigin おける数値実験の結果

	次元数	手法	平均値	最良値	最悪値	標準偏差
Rastrigin	50	CS	452.25	383.40	524.08	35.66
		PR	367.02	295.15	470.92	41.08
		ACS1	354.28	241.71	437.77	47.63
		ACS2	356.31	269.15	499.67	45.54
	100	CS	1197	1085	1316	52.6
		PR	909	754	1052	71.1
		ACS1	891	725	1021	64.6
		ACS2	899	726	1044	67
	300	CS	4533	4313	4780	98.9
		PR	3726	3391	4007	138.7
		ACS1	3684	3307	3932	124.1
		ACS2	3678	3410	3872	108
	1000	CS	17031	16703	17468	159
		PR	15126	14517	15740	157
		ACS1	15111	14686	15739	227
		ACS2	15076	14494	15543	239

表 4.3 : Levy おける数値実験の結果

	次元数	手法	平均値	最良値	最悪値	標準偏差
Levy	50	CS	40.73	10.85	63.18	10.70
		PR	29	19.41	41.10	4.99
		ACS1	27.60	19.45	36.48	4.17
		ACS2	27.84	18.75	41.29	4.87
	100	CS	67.98	55.76	87.45	6.56
		PR	39.61	32.02	49.02	4.27
		ACS1	39.63	27.26	57.38	5.05
		ACS2	38.87	29.85	47.15	3.90
	300	CS	112.00	97.35	122.26	6.19
		PR	71.55	60.71	82.19	4.82
		ACS1	70.01	60.43	80.43	4.61
		ACS2	70.30	62.47	77.91	4.04
	1000	CS	147.12	139.55	155.15	3.40
		PR	111.80	104.18	121.06	4.21
		ACS1	110.85	102.49	122.40	4.66
		ACS2	110.67	100.45	122.02	4.18

表 4.4 : Ackley おける数値実験の結果

	次元数	手法	平均値	最良値	最悪値	標準偏差
Ackley	50	CS	2.96	2.55	3.94	0.25
		PR	6.02	2.83	7.94	1.32
		ACS1	5.55	2.79	7.70	1.24
		ACS2	5.06	2.69	8.13	1.33
	100	CS	4.72	4.21	5.39	0.26
		PR	7.56	5.68	8.71	0.71
		ACS1	7.23	5.31	8.65	0.75
		ACS2	7.25	5.33	8.53	0.86
	300	CS	8.32	7.85	8.98	0.24
		PR	9.11	8.70	9.43	0.15
		ACS1	9.06	8.27	9.39	0.22
		ACS2	9.03	8.50	9.39	0.22
	1000	CS	10.15	10.01	10.28	0.06
		PR	9.89	9.74	10.05	0.06
		ACS1	9.85	9.71	10.01	0.06
		ACS2	9.85	9.68	9.99	0.07

表 4.5 : Griewank おける数値実験の結果

	次元数	手法	平均値	最良値	最悪値	標準偏差
Griewank	50	CS	1.08	1.05	1.11	0.01
		PR	0.86	0.66	1.01	0.08
		ACS1	0.67	0.46	0.83	0.09
		ACS2	0.67	0.46	0.89	0.09
	100	CS	1.89	1.68	2.13	0.10
		PR	1.52	1.33	1.82	0.11
		ACS1	1.25	1.16	1.41	0.05
		ACS2	1.26	1.17	1.36	0.04
	300	CS	22.76	19.12	25.01	1.27
		PR	18.60	15.98	21.58	1.25
		ACS1	14.87	12.77	17.14	1.04
		ACS2	14.35	12.46	16.91	0.98
	1000	CS	182.04	174.06	190.42	3.55
		PR	149.60	138.51	157.16	4.35
		ACS1	135.51	127.34	142.82	3.04
		ACS2	134.62	126.57	143.43	3.61

表 4.6 : Alpine おける数値実験の結果

	次元数	手法	平均値	最良値	最悪値	標準偏差
Alpine	50	CS	47.63	37.77	58.82	5.33
		PR	30.40	18.98	42.50	5.03
		ACS1	29.56	19.05	45.48	6.04
		ACS2	27.65	18.03	37.13	4.38
	100	CS	150.74	119.96	178.66	13.12
		PR	102.36	80.90	132.20	11.23
		ACS1	94.91	75.49	112.86	9.55
		ACS2	94.10	76.22	119.82	9.86
	300	CS	666.68	613.75	729.11	23.15
		PR	511.84	453.96	563.28	28.27
		ACS1	501.44	439.74	569.07	29.00
		ACS2	501.47	439.45	596.92	30.60
	1000	CS	2728	2639	2829	411
		PR	2330	2191	2423	526
		ACS1	2299	2210	2391	441
		ACS2	2316	2188	2424	49

5

結論

本研究では，空間的な多様化・集中化の先行研究における1つの課題を解決し，時空間の多様化・集中化の適応型 Cuckoo Search を提案手法した。高いレベルの時間的な多様化・集中化を実現し，CS アルゴリズムの探索性能を向上した。さらに，時空間の多様化・集中化の戦略を使って新たな排斥手法も提案した。

5.1 まとめ

本研究は先行研究の Cuckoo Search の上で高いレベルの時間的な多様化・集中化を実現した。具体的には，本研究で提案した新たな CS は，探索序盤では広い範囲で探索することで多様化を実現して，長期的な解の改善を実現した一方で，探索終盤では良い解付近を集中的に探索することで短期的な解の改善を実現した。提案した「時空間の多様化・集中化に基づく適応型 Cuckoo Search」では，図 5.1 のように探索のイテレーション回数に応じて多様化・集中化が十分に実現できる。十分な時間的な多様化・集中化を実現するために，今回パラメータ β の上下限がスケジュールとして設定されており，パラメータ β の上下限を調整することで多様化と集中化の上下限を制御する。さらに，この制御はイテレーション回数の変化に応じて適応的に実現される。言い換えれば，探索のイテレーション回数，ベンチマーク関数の種類，次元数，初期値などに依らずに，探索初期の十分な多様化と探索終盤の十分な集中化を確実に保証できることを意味し，本研究を通じて先行研究の CS

の近傍生成の時間的な多様化・集中化が不十分であるという課題を解決したことになる。

以上，本研究は空間的な多様化・集中化を実現するために，各探索点の評価値に応じてランキングに基づきパラメータを調整する戦略と，時間的な多様化・集中化戦略を有機的に結合した新たな適応型 CS を提案した。さらに，提案した近傍生成と同じ探索戦略を用いた新たな排斥手法も提案した。そして，典型的なベンチマーク問題を用いた数値実験を通じて，提案手法の有用性を検証した。

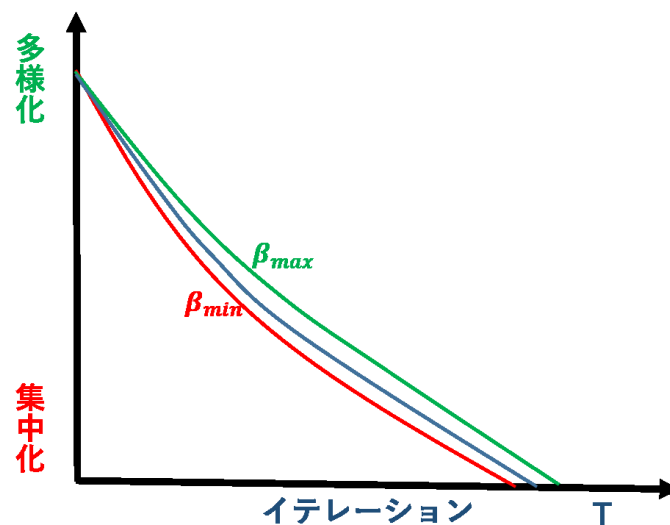


図 5.1：提案手法の多様化と集中化の程度

以下では，改めて本論文の内容をまとめる。

本論文ではまずメタヒューリスティクスの発展と代表的なメタヒューリスティクスについて紹介した。そして，先行研究と本研究では一般的なメタヒューリスティクスと比べて，特徴がある Cuckoo Search というアルゴリズムに着目した。Cuckoo Search の特徴は他の個体の情報だけ用いて近傍解を生成することと，レヴィ分布によって近傍生成し，パラメータにより近傍生成の範囲を調整している。この Cuckoo Search の特別な特徴を活かして，探索性能を向上することが考えられる。先行研究では Cuckoo Search の特徴の上で空間的な多様化・集中化と近接最適性原理（POP）を使って，Cuckoo Search のパラメータの適応化を実現した。先行研究では Cuckoo Search の探索性能が向上したが，時間的な多様化・

集中化が不十分であるという課題が残っており、本研究ではこれを解決することで性能向上を目指した。先行研究の上で時間的な多様化・集中化の戦略を使って、先行研究の課題を解決し、高いレベルで多様化・集中化を確実に実現した。さらに、Cuckoo Search の排斥操作に時間的な多様化・集中化の戦略を使ってパラメータの自由度を保持し、新たな排斥手法も提案した。排斥操作の適応性も実現した。最後に数値実験を通じて、多くのベンチマーク問題において従来の手法よりも、提案手法の性能が高いことを検証した。

5.2 展望

本研究は先行研究の時間的な多様化・集中化が不十分であるという課題を解決し、時空間的な多様化・集中化の Cuckoo Search を提案した。さらに、Cuckoo Search の排斥操作にも同じの探索戦略を使って、新たな排斥手法も提案した。しかし、本研究は排斥操作を追加した後に、高次元（1000 次元）問題においてはほとんどケースにおいて結果が優れているが、低次元問題においては提案手法の結果は先行研究より悪化する場合もある。今後は、(1) 近傍生成と排斥操作のパラメータの調整方法に性能を高め、近傍生成と排斥操作のパラメータの調整方法の協調性を高めることで、低次元と高次元の双方における探索性向上が課題、および (2) 排斥の機能に着目し、さらなる排斥の性能を向上させることが、課題として挙げられる。

参考文献

参考文献

- [1] 安田恵一郎：「メタヒューリスティクスの現在と未来」，計測と制御，Vol.47，No.6，pp.453–458（2008）
- [2] 玉置久 編著：「システム最適化」，オーム社（2005）
- [3] 田邊遼司・串田準一・畠中利治：「関数最適化における進化計算」，計測と制御，Vol.54，No.8，pp.567–572（2015）
- [4] 篠崎隆宏・渡邊晋治：「音声認識とブラックボックス最適化」，日本音響学会誌，Vol.72，No.10，pp.644–652（2016）
- [5] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” Proceedings of IEEE International Conference on Neural Networks, Vol.4, pp.1942–1948 (1995)
- [6] R. M. Storn and K. V. Price: “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Space,” Journal of Global Optimization, Vol.11, No.4, pp.341–359 (1997)
- [7] (社)電気学会 進化技術応用調査専門委員会：「進化計算ハンドブック＜第Ⅰ巻＞基礎編」，近代科学社（2010）
- [8] X. S. Yang: “Nature-Inspired Metaheuristic Algorithms,” Second Edition, Luniver Press (2010)

- [9] X. S. Yang and S. Deb: “Cuckoo Search via Lévy Flights,” Proceedings of World Congress on Nature and Biologically Inspired Computing 2009, pp.210–214 (2009)
- [10] 相吉英太郎・安田恵一郎 編著：「メタヒューリスティクスと応用」，電気学会，オーム社（2007）
- [11] R. N. Mantegna: “Fast, Accurate Algorithm for Numerical Simulation of Lévy Stable Stochastic Processes,” Physical Review E, Vol.49, No.5, pp.4677–4683 (1994)
- [12] 熊谷，田村，土屋，安田：「探索状態の評価と制御に基づく適応型 Cuckoo Search」，電気学会 電子・情報・システム部門誌, Vol.136, No.11, pp.1596–1609（2016）
- [13] Y. Miyake, K. Tamura, J. Tsuchiya, and K. Yasuda, “Adaptive Cuckoo Search Based on Ranking of Search Point,” Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp.1788–1792 (2017)
- [14] Y. Miyake, W. Kumagai, K. Tamura, and K. Yasuda, “Search Point Ranking Based Adaptive Cuckoo Search,” IEEJ Transactions on Electrical and Electronic Engineering, Vol.13, No.7, pp.1075–1076 (2017)
- [15] 曹，田村，土屋，安田：「多様化と集中化の制御に基づく適応型 Cuckoo Search」，計測自動制御学会 システム・情報部門 学術講演会 2018, GS01-07（2018.11）

A

ベンチマーク関数

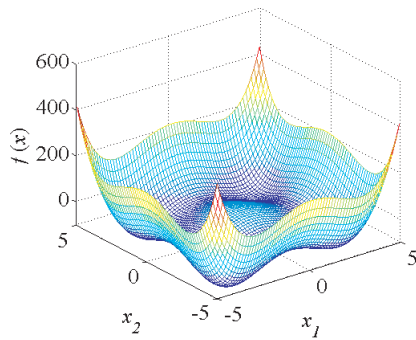
本論文で用いたベンチマーク関数について記述する。

1. 2^n minima Function

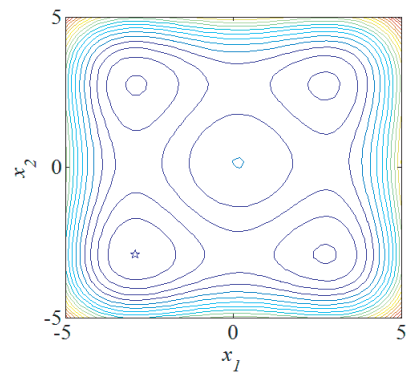
$$f_3(\boldsymbol{x}) = \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i) \quad (\text{A.1})$$

$$\boldsymbol{x}^o \approx (-2.92, -2.92, \dots, -2.92), \quad f(\boldsymbol{x}^o) \approx -78n$$

$$\boldsymbol{S} = [-5 \ 5]^n$$



(a) 概形



(b) 等高線

図 A.1 : 2^n minima Function

2. Rastrigin Function

$$f_4(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (\text{A.2})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

$$S = [-5 \ 5]^n$$

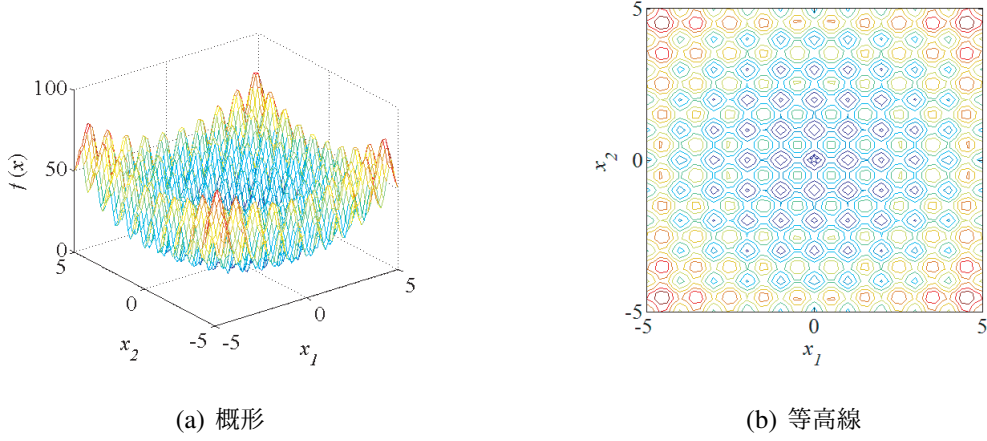


図 A.2 : Rastrigin Function

3. Lévy Function

$$f_6(\mathbf{x}) = \frac{\pi}{n} \left(\sum_{i=1}^{n-1} \{(x_i - 1)^2 (1 + 10 \sin^2(\pi x_{i+1}))\} + 10 \sin^2(\pi x_1) + (x_n - 1)^2 \right) \quad (\text{A.3})$$

$$\mathbf{x}^o = (1, 1, \dots, 1), \quad f(\mathbf{x}^o) = 0$$

$$S = [-5 \ 5]^n$$

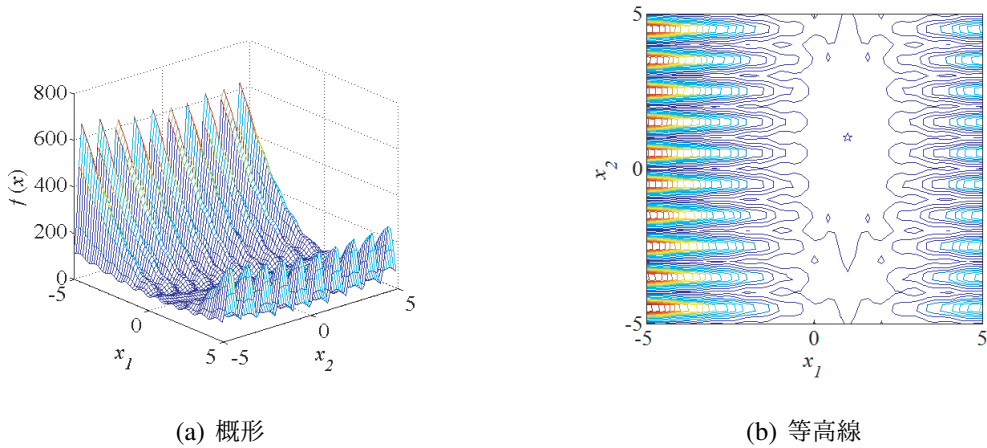


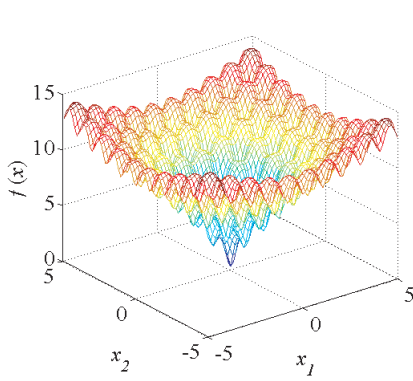
図 A.3 : Lévy Function

4. Ackley Function

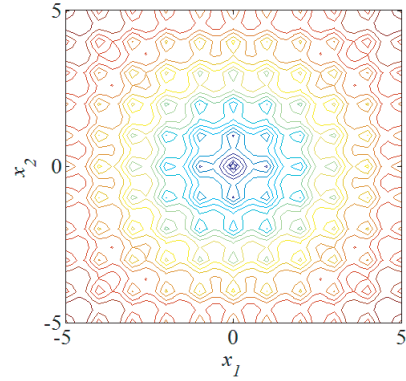
$$f_7(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + \exp(1) \quad (\text{A.4})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

$$S = [-5 \ 5]^n$$



(a) 概形



(b) 等高線

図 A.4 : Ackley Function

5. Griewank Function

$$f_8(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n \left\{ x_i + \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1 \right\} \quad (\text{A.5})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

$$S = [-50 \ 50]^n$$

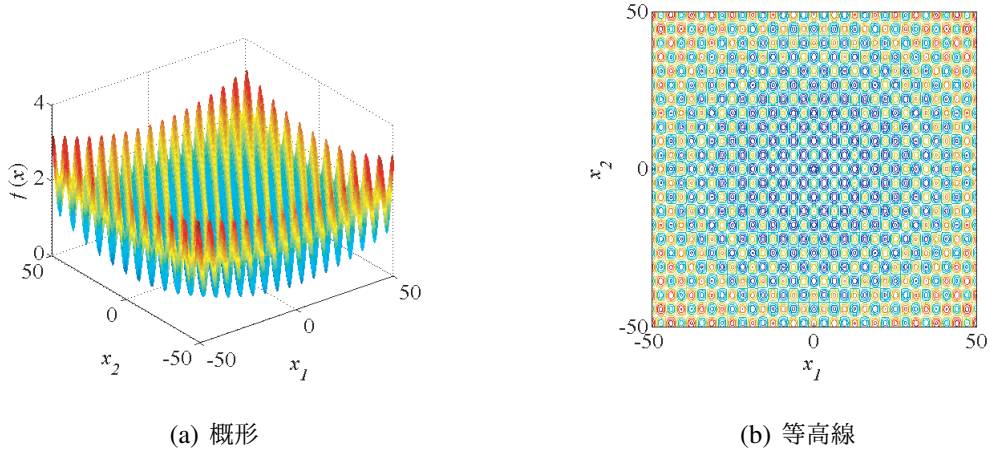


図 A.5 : Griewank Function

6. Alpine Function

$$f_9(\mathbf{x}) = \sum_{i=1}^n |x_i \sin x_i + 0.1x_i| \quad (\text{A.6})$$

$$\mathbf{x}^o = (0, 0, \dots, 0), \quad f(\mathbf{x}^o) = 0$$

$$\mathbf{S} = [-10 \ 10]^n$$

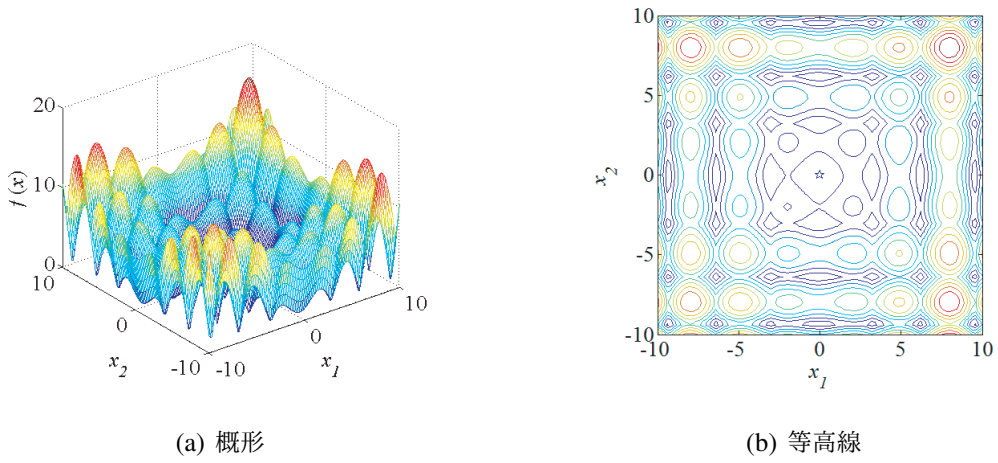


図 A.6 : Alpine Function

謝辞

本論文は、著者が首都大学東京大学院 システムデザイン研究科 博士前期課程において、首都大学東京大学院 システムデザイン研究科 電子情報システム工学域 安田 恵一郎 教授の指導の下で行ったメタヒューリスティクスに関する研究成果である。

本研究の遂行および本論文の作成にあたり、日頃からご指導頂いている安田 恵一郎 先生をはじめ、助教 土屋 淳一 先生、田村 健一 先生、システム制御工学研究室の方々には、多くの御指導、御助言を頂き、誠にありがとうございました。